General Disclaimer

One or more of the Following Statements may affect this Document

- This document has been reproduced from the best copy furnished by the organizational source. It is being released in the interest of making available as much information as possible.

- This document may contain data, which exceeds the sheet parameters. It was furnished in this condition by the organizational source and is the best copy available.

- This document may contain tone-on-tone or color graphs, charts and/or pictures, which have been reproduced in black and white.

- This document is paginated as submitted by the original source.

- Portions of this document are not fully legible due to the historical nature of some of the material. However, it is the best reproduction available from the original submission.

(NASA-CR-147753) LACIE PERFORMANCE PREDICTOR FINAL OPERATIONAL CAPABILITY PROGRAM DESCRIPTION, VOLUME 1 (TRW SYSTEMS GROUP) 339 p HC $10.00 CSCL 02C
N76-25630
UNCLAS
G3/43 42116

28234-6027-RU-00

NASA CR-147753

LACIE PERFORMANCE PREDICTOR FINAL OPERATIONAL CAPABILITY PROGRAM DESCRIPTION

VOLUME I

MAY 1976

Prepared for

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION
Lyndon B. Johnson Space Center
Houston, Texas

Contract Number NAS-9-14547

TRW
SYSTEMS GROUP
ONE SPACE PARK · REDONDO BEACH · CALIFORNIA 90278

This document, in three volumes, describes the FOC version of the LACIE Performance Predictor produced under Contract NAS9-14547. NASA/JSC Contract Technical Monitor is I. D. Browne, Earth Observations Division.

Table of Contents

EPHEMS BOOK I

Table of Contents

PART I

PROBLEM DESCRIPTION

PROBLEM DESCRIPTION FOR EPHEM PROGRAM

1.0 SCOPE

1.1 PROGRAM CAPABILITIES

This program will compute the orbital parameters for up to 2 vehicles orbiting about the earth for up to 549 days. This data is sorted on disk files and represents a continuous swath path about the earth. These swath tables are to be used by the segment access generator (SAGE) program to determine when and if certain target land segments lie in the path of the swath.

1.2 METHOD OF PROGRAM DEVELOPMENT

Use will be made of existing FORTRAN subroutines in order to compute the ephemeris data and the data for the swath tables. Except for a limited number of specific places, the equation processing portion of these routines will remain intact. Control logic will be altered but new routines will only have to be created for processing card input and writing the swath files.

1.3 OPERATIONAL ASSUMPTIONS

- There can be a maximum of 2 vehicles.
- There can be a maximum of 549 days.
- The latitude band for swath table generation is $\pm 65^{\circ}$ latitude and there will be a maximum of 100 latitude points to process.
- Only 1 case is run at a time.
- The complete card data set must be entered for each run.
- The existing swath tables will be destroyed each time this program is run.
- The swath tables will be designated as permanent files since it is assumed that this program will be run about once per year.
- Since namelist is not available on the PDP, fixed field card formats are utilized.

- A sun synchronous inclination will be computed for each run condition
- Only first order oblateness terms are modeled in the orbit generation, i. e., nodal regression and apsidal precession.
- Orbital elements are maintained as Keplerian elements (a, e, i, Ω, w, t) and are updated on a rev by rev basis.
- Orbit initialization equations will insure passage through a reference point at a specified time on the first day. Active passage is assumed to be on the descending pass.

2.0 INPUT

The only input media to this program are cards. The following sections give the list and definitions of the input quantities, the card formats and rules of entry.

2.1 LIST OF INPUT QUANTITIES

See input data description sheets on pages 6 and 7.

2.2 CARD FORMATS

See card format sheet on page 8.

2.3 DECK SETUP

The order of input is:

1. Header card
2. Program control card
3. Orbital determination data (2 cards)

2.4 RULES FOR ENTERING DATA ON CARDS

2.4.1 General

1. Integers must be entered right justified.
2. F-format numbers must have the decimal point present, i.e., F5.1-XXX.X, F4.0-XXX.
3. The card sequence numbers in the C.C. 79-80 must be present in all data cards.

2.4.2 Rules for Specific Fields

IVEH - No more than 2 vehicles can be entered in this field.

NODAY - A number larger than 549 will not be accepted.

IYR - An entry less than 64 will not be accepted.

INLAT and ISLAT - A negative entry or an entry greater than 65 will not be accepted. The total number of latitudes allowed, represented by the combined bands, is 100.

IPPI5 - An entry larger than 549 will not be accepted.

SA - An absolute value greater than 10° is not allowed.

IV1TIM and IV2TIM - The first subscript entry (year) must be between 1964 and the present.

Input Data Description

Program Control Input

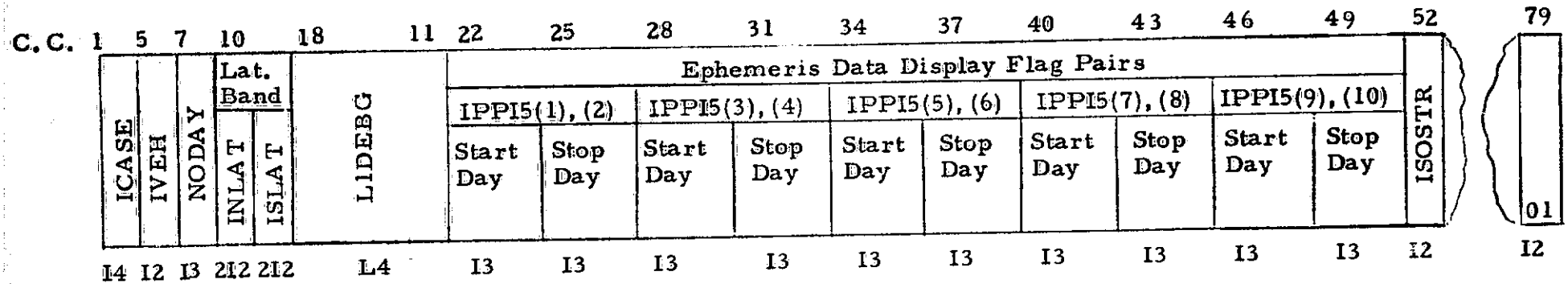
Name	Symbol	Dimension	Nominal Value	Range	Units	Description
ICASE		1	0	1-9999	--	A 4 digit case number to identify the printed output and the swath tables
NODAY		1	549	1-549	--	No. of days to generate ephemeris data for the swath table
IVEH		1	2	1-2	--	Number of vehicles to process
HEADER		8	Blanks	---	--	Provision for an 72 character case header to print out at the top of every output print page
LIDEBG		4	F (Blank)	T or F	--	Flag to allow printout of intermediate data for swath calculations. Data for the following rouitnes is printed based on subscript values: 1. SWATH 2. Swath table and 4. REVTAB reference file
INLAT		2	5, 65	0-65	Deg	Northern Hemisphere latitude band for swath generation
ISLAT		2	15, 45	0-65	Deg	Southern Hemisphere latitude band for swath generation
IPPI5(10)		10	1, 549 8*0	1-549	--	Ephemeris data display flag = [start day, end day] pairs
ISOSTR		1	6	0-12	--	No. of months delay to start the generation of swath data on the swath table for the southern latitudes

Input Da. Description

Orbital Determination Input

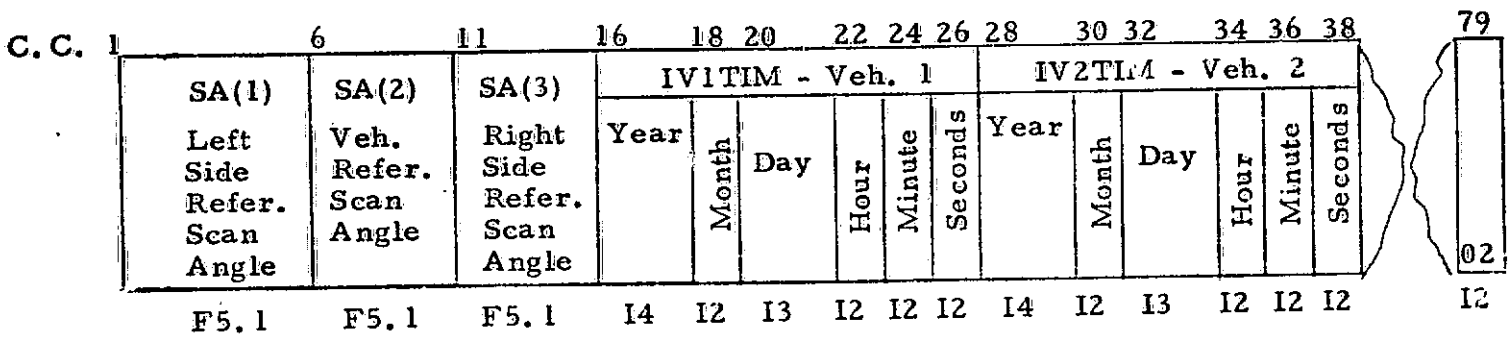
Name	Symbol	Dimension	Nominal Value	Range	Units	Description
IV1TIM		6	0	→	→	Vehicle 1 reference latitude passage time IV1TIM(1)-(Year-1900); IV1TIM(2)-Month r IV1TIM(3)-Day No.; IV1TIM(4)-Hours IV1TIM(5)-Minutes; IV1TIM(6)-Seconds
IV2TIM		6	0	→	→	Vehicle 2 reference latitude passage time; same information as Vehicle 1.
ORBIT1		6	0			Orbital elements for Vehicle 1
(1)	a			6650 to 7700	Kilom.	Semi-major axis
(2)	e			10^{-5}<.15	--	Eccentricity
(4)	Ω			0-360	Deg.	Longitude of Reference Latitude
(5)	ω			0-360	Deg.	Argument of perigee
(6)	λ			0- ±65°	Deg.	Vehicle reference latitude
ORBIT2		6	0			Orbital elements for Vehicle 2. Same data as for Vehicle 1.
SA		3	0	-10-+10	Deg.	Reference scan angles defining swath field of view

PROGRAM CONTROL CARD FORMAT



ORBITAL DETERMINATION DATA CARD FORMATS

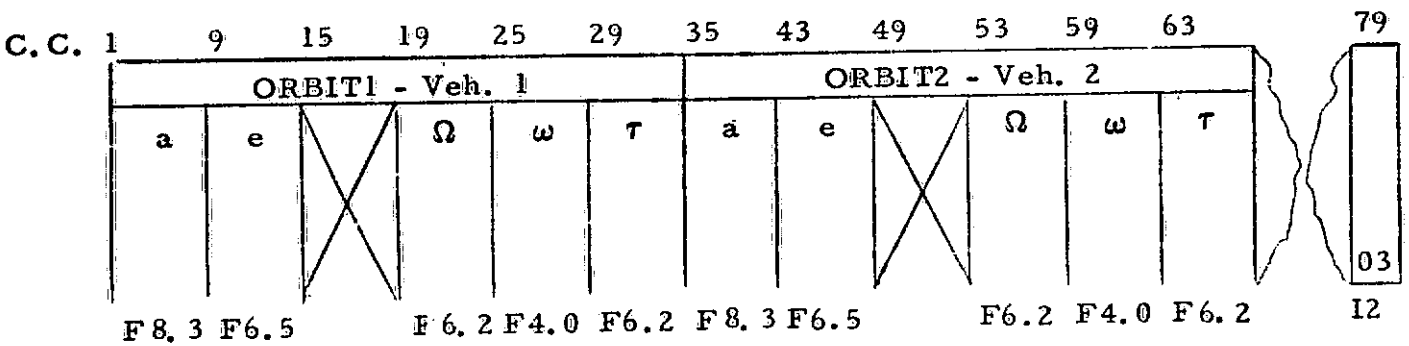
CARD 1



Data Types

- I - Integer
- AN - Alphanumeric
- F - Floating Decimal
- L - Logical (T or F)

CARD 2



3.0 PROCESSING

3.1 OVERVIEW

An existing analytic ephemeris program is being used to simulate the orbital process which maintains the data as follows:

(1) A set of Keplerian orbital elements for each rev (a, e, i, Ω, ω, t).

(2) Ground path and swath visibility tables as a function of latitude to be used by SAGE in determining access conditions.

The only original code for this program will be related to control logic and orbit initialization procedures.

3.2 PROGRAM FLOW

The subroutine MAIN will be used as the master control for this program. Its logic flow and functions are shown in the MAIN block diagram on Page 11. This diagram illustrates the usage of the major subroutines. The methodology for the generation of ephemeris data is performed in HECTOR and its flow is shown on pages and . Subroutine ACCGEN controls the computation of swath data for one revolution. The list of major routines performing mathematical calculations are:

1. HECTOR (Called by MAIN)
   Computes for each revolution:
   - Semi-major axis
   - Eccentricity
   - Inclination
   - Longitude of the node
   - Argument of perigee
   - Time of perigee passage
   - Start time of rev
   - Right ascension of the node

2. LFPA - entry ALPHA (Called by HECTOR).
   Computes right ascension of true and mean sun.

3. KEPLER (Called by HECTOR)
   Computes the eccentric anomaly by iteration technique, i.e., solves for E in $\{ E - e \sin E \equiv n \Delta t \}$

4. ACCGEN
   Computes eccentric anomaly at ascending node - E. Call SWATH and REVTAB to generate swath data for one revolution.

5. SWATH
   Generates the latitude and delta longitude data for the scan angles representing a swath path about the earth as a function of eccentric anomaly.

6. REVTAB
   Reformats the data from the SWATH routine data to represent this data at discrete 1° latitude increments.

3.2.1 Initialization Procedures to Ensure a Sun Synchronous and Reference Passage Condition (LACIE Requirement)

(a) The input inclination is overriden in INIT by a computed inclination which ensures a sun synchronous orbit. A sun synchronous orbit is defined as an orbit whose nodal regression rate matches the mean sun motion per day:

Recompute i ≡ (ORBIT1(3) and ORBIT2(3) to make the orbit sun synchronous.

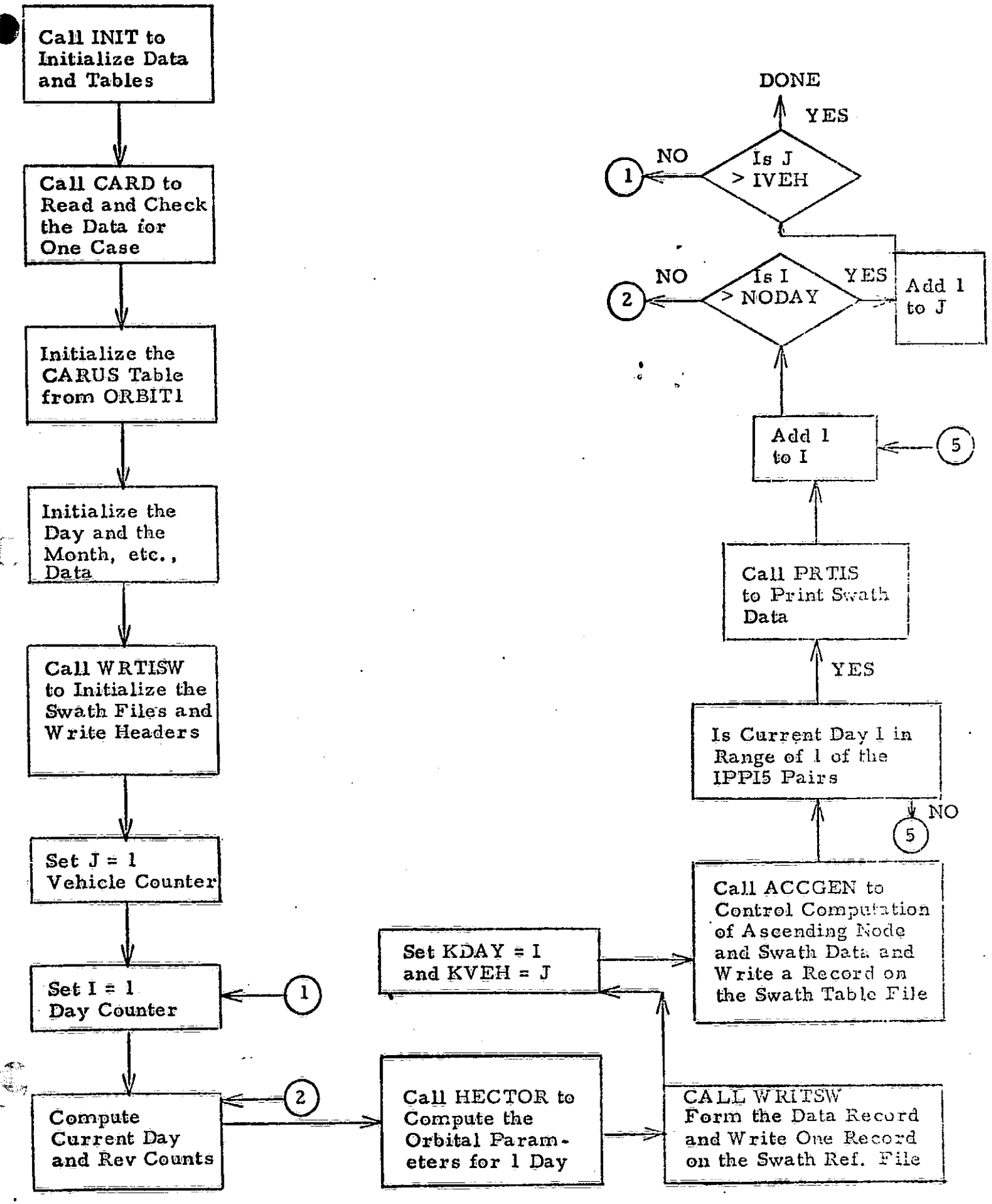
$$i = \cos^{-1} \left[\frac{-4\pi (a^2)(1 - e^2)^2}{3 * (CMU/a^3)^{1/2} * CJ2X365.25*86400} \right] \quad (1)$$

(b) An initial rev node, Ω node, and time of node, t node, will be computed from the input (λ, Ø) [reference point] and passage time IV1TIM, IV2TIM) for each active vehicle. The procedure is:

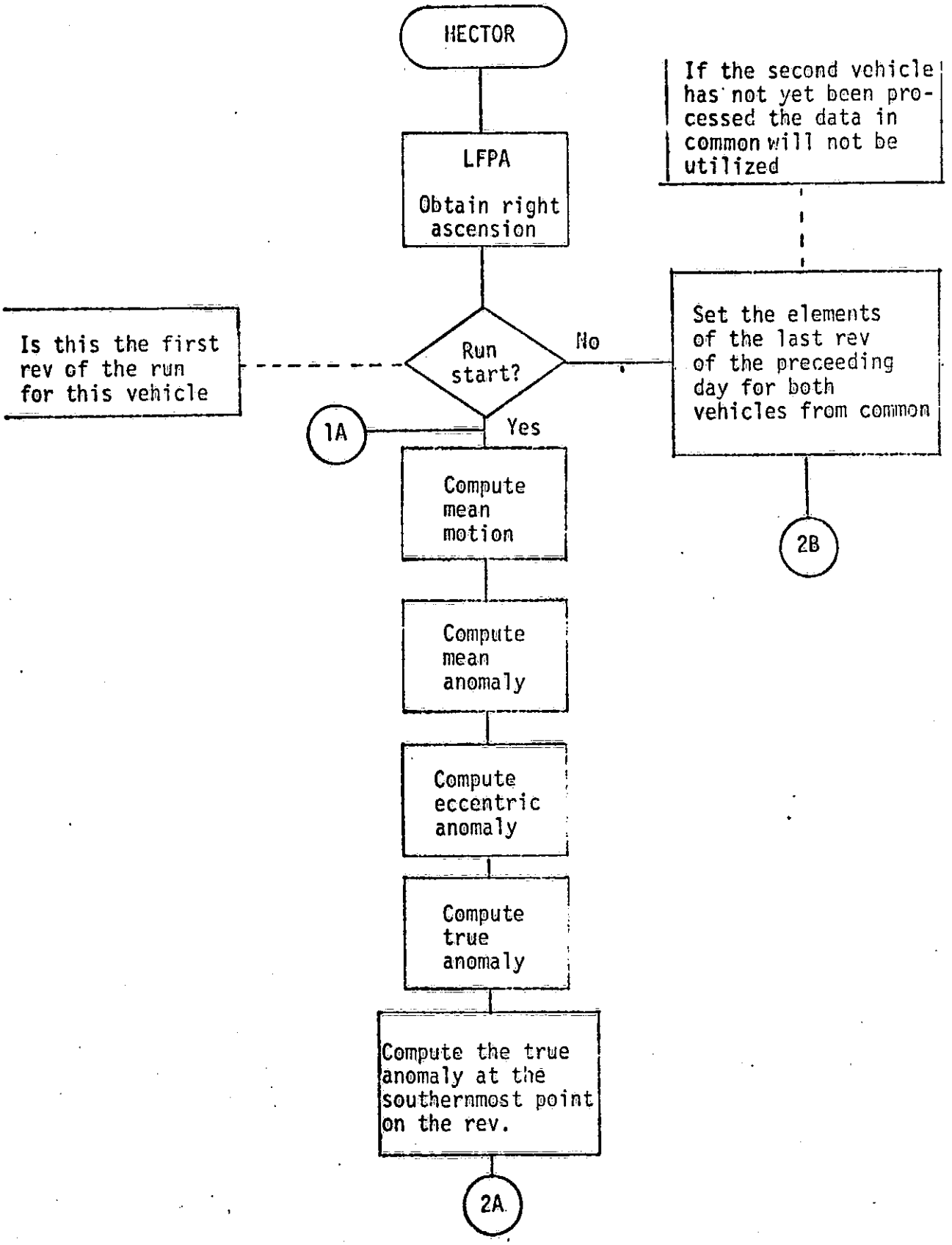
(1) Back off K1, K2 revs such that IV1TIM, IV2TIM remain within current day.

Program Flow for Ephemeris Program

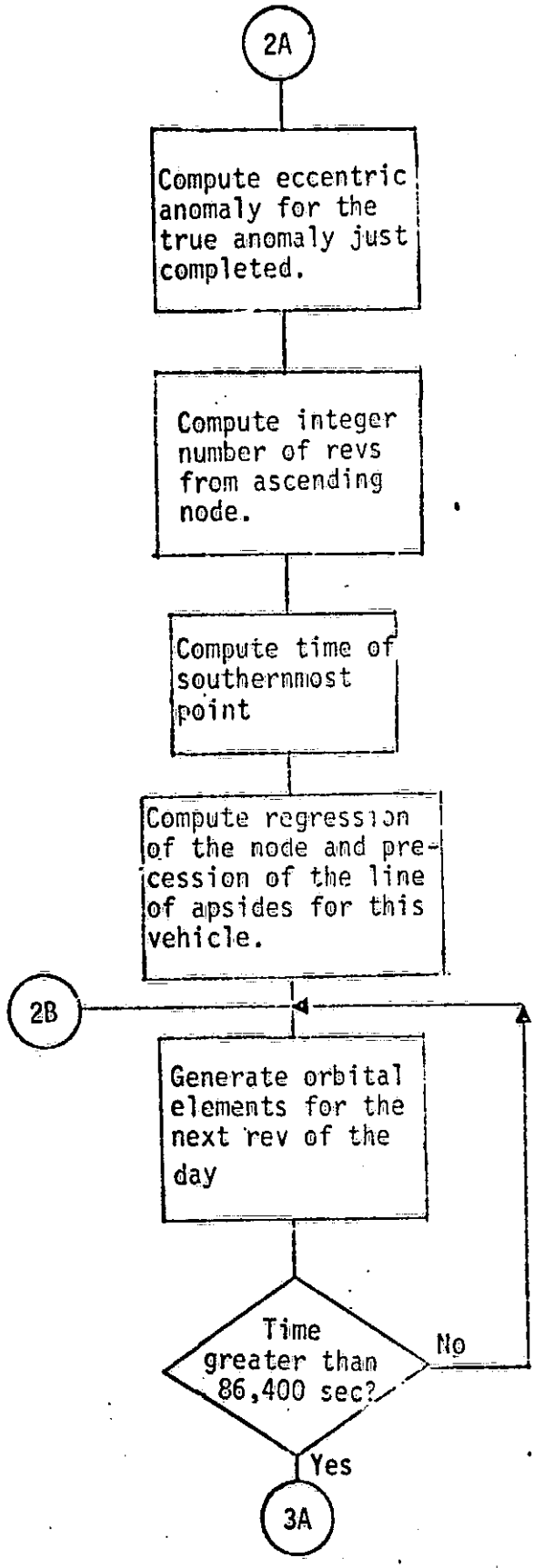
Master Control - MAIN



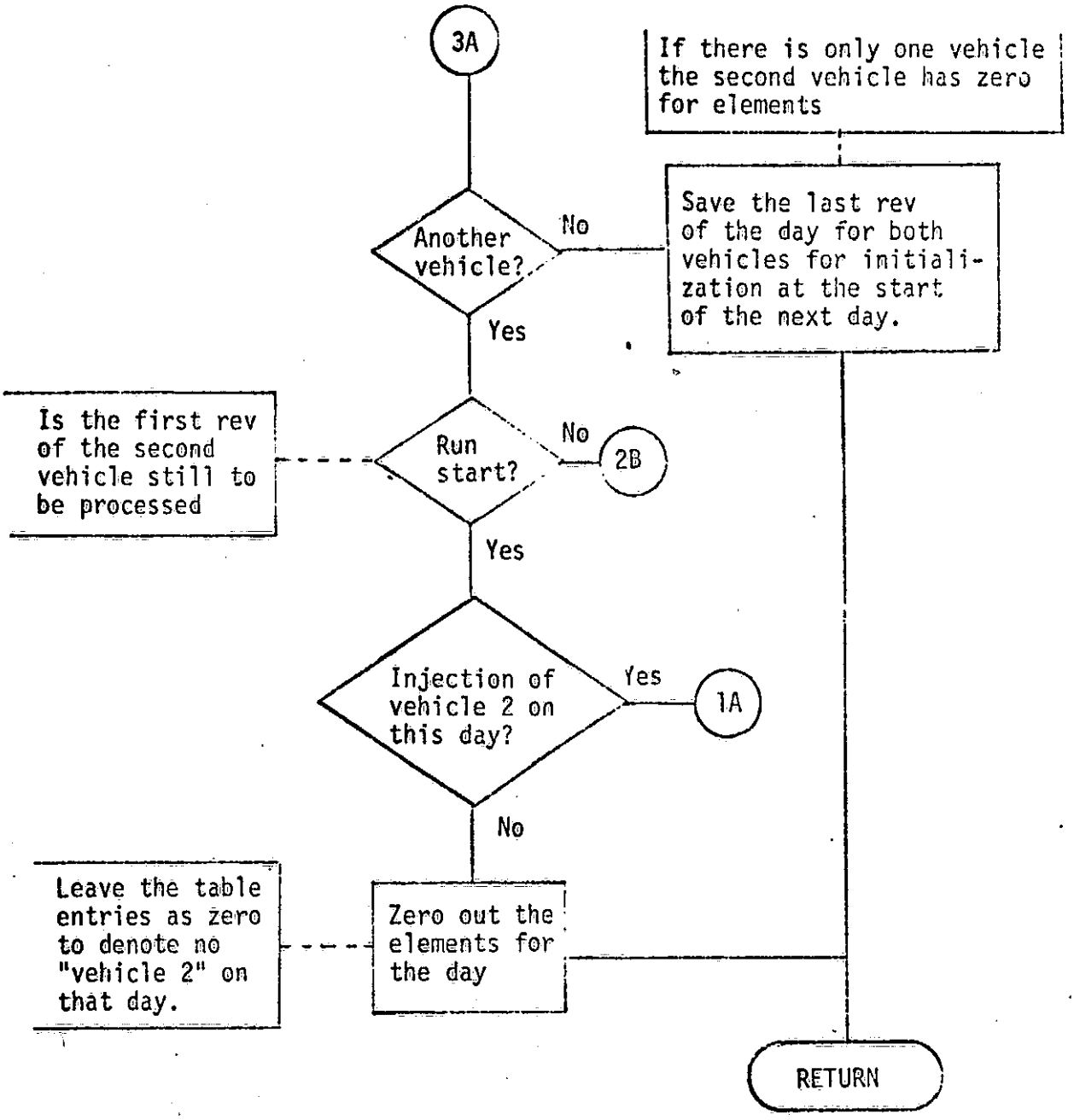
EPHEMERIS FLOW



EPHEMERIS FLOW (CONT'D)



EPHEMERIS FLOW (CONT'D)



(2) Compute initial rev node by

$$\Omega_{node} = \phi_{input} + \pi - \sin^{-1}\left(\frac{-\tan\lambda}{\tan i}\right) + K_i \Delta\Omega_{rev} \qquad (2)$$

where,

$$\Delta\Omega_{rev} = \frac{4\pi^2}{86400\,\bar{n}}$$

$$\bar{n} = n_o\left[1 + \frac{3}{2} J_2 \sqrt{\frac{1-e^2}{p^2}}\left(1 - \frac{3}{2}\sin^2 i\right)\right]$$

$$n_o = \sqrt{Mo/a^3}$$

= mean motion

$$p = a(1 - e^2)$$

μ_o = gravitational constant

a = semi-major axis

These items were incorporated into the HECTOR initialization procedure. The mean motion calculation, $\bar{n}$, was implemented by perturbing μ calculation.

3.2.2 Computation of Orbital Elements for Initial Rev of Simulation

HECTOR will obtain the right ascension of Greenwich for the initial rev of the day. At the start of the run HECTOR will utilize the injection conditions for each vehicle to compute the elements at the start of the rev. The following equations will be utilized for computation of the elements of the first rev of the simulation.

$$\mu = \mu_o K^2$$

where,

$$K = 1 + \frac{3}{2} J_2 \sqrt{\frac{1-e^2}{p^2}}\left(1 - \frac{3}{2}\sin^2 i\right) \qquad (3)$$

$$\text{Mean Motion} = n = \sqrt{\mu/a^3} \tag{4}$$

$$\text{Mean Anomaly} = n\ (t_o - T) = M \tag{5}$$

where

a ≡ semi-major axis

t_o ≡ time of injection in seconds from midnight

T ≡ time of perigee passage

and

$t_o > T$ (if $T > t_o$ then $T = T - \frac{2\pi}{n}$)

The eccentric anomaly, E, is defined by

$$E - e \sin E = M \tag{6}$$

and the solution for E is obtained from the following iterative technique with ϵ defined as the required accuracy for E.

(a) assume E_o and compute M_o

(b) compute $\Delta M = M - M_o = n(t_o - T) - M_o$

(c) compute $1 - e \cos E_o$

(d) $\Delta E = \Delta M / (1 - e \cos E_o)$

(e) $E_1 = E_o + \Delta E$

(f) $M_1 = E_1 - e \sin E_1$

(g) $\delta M = M - M_1$

(h) if $\delta M > \epsilon$ go to (c) with $E_o = E_1$ and $\Delta M = \delta M$

The true anomaly at injection is

$$\nu = \sin^{-1}\left(\frac{\sqrt{1-e^2}\,\sin E}{1-e\cos E}\right) \tag{7}$$

and the quadrant of ν is determined from Equation (7) and that $\cos E - e < 0$ if ν is in the 2nd or 3rd quadrant since

$$\cos\nu = \frac{\cos E - e}{1 - e\cos E}.$$

The inplane angle from the ascending node to the injection point is given by

$$u = (\omega + \nu)_{\text{MOD } 2\pi} \tag{8}$$

where ω is the argument of perigee from the input.

The inplane angle from the injection point to the southernmost point on the rev (i.e. the start of the next rev) is

$$\Delta\nu = \frac{3}{2}\pi - u \tag{9}$$

and the true anomaly at the southernmost point is

$$\nu_1 = (\nu + \Delta\nu)_{\text{MOD } 2\pi} \tag{10}$$

The cosine of the eccentric anomaly at ν, is

$$\cos E_1 = \frac{1}{e}\left(1 - \frac{1-e^2}{1+e\cos\nu_1}\right) = \frac{e + \cos V_1}{1 + e\cos V_1} \tag{11}$$

The eccentric anomaly at ν_1 is

$$E_1 = \sin^{-1}\left[\frac{\sin\nu_1(1-e\cos E_1)}{\sqrt{1-e^2}}\right] \tag{12}$$

with the quadrant of E_1 uniquely determined by (11) and (12).

The integer number of perigee to perigee revs prior to the southernmost point is given by

$$k = [(\nu + \Delta\nu) / 2\pi]_{int} \tag{13}$$

The time at the sounthernmost point is

$$T_s = \frac{E_1 - e\sin E_1}{n} + T + k\left(\frac{2\pi}{n}\right) \tag{14}$$

where $n \equiv$ mean motion

$T \equiv$ time of perigee passage

Likewise, the time of the ascending node for the rev is

$$T_1 = \frac{E^* - e\sin E^*}{n} + T + k\left(\frac{2\pi}{n}\right) \tag{15}$$

where $E^* = E_1 + \pi/2$

The regression of the node and precession of the line of apsides are computed from

$$\dot{\Omega} = \frac{-3}{2} \frac{n J_2}{a^2(1-e^2)^2} \cos i \tag{16}$$

$$\dot{\omega} = \frac{-3}{2} \frac{n J_2}{a^2(1-e^2)^2} \left(\frac{5 \sin^2 i}{2} - 2 \right) \tag{17}$$

where
- a ≡ semi major axis
- e ≡ eccentricity
- n ≡ mean motion
- J_2 ≡ second harmonic of potential function

If the inclination of the orbit is greater than $\pi/2$ the node does not undergo regression but advances as shown by Equation (16).

From equations (15), (16), and (17) the longitude of the node and the argument of perigee at the time of the node of the first rev can be obtained.

$$\Omega_1 = \Omega_o + (\dot{\Omega} - \omega_e)(T_1 - t_o) \tag{18}$$

$$\omega_1 = \omega_o + \dot{\omega}(T_1 - t_o) \tag{19}$$

where
- ω_e ≡ the rotation rate of the earth and is a positive value (rad/sec)
- t_o ≡ the time of injection

The time of perigee passage for the first rev is obtained from

$$\tau_1 = T + (k+1)\,(2\pi/n) \tag{20}$$

where

T ≡ the input time of perigee passage

k ≡ value from equation (13)

The right ascension of the node at the time of nodal crossing is given by

$$\lambda_1 = [\alpha_{g_o} + \omega_e T_1 + \Omega_1]_{\text{MOD } 2\pi} \tag{21}$$

where

α_{g_o} ≡ the right ascension of Greenwich at midnight from the subroutine LFPA

The angle between the sun and the orbit plane will be evaluated for the first rev of each day by differencing the right ascension of the ascending node and the right ascension of the sun. The right ascension of the sun employs many of the same equations just presented except the sun is the central body and the earth is the "vehicle". These calculations are performed by LFPA. From Equations (5) and (6) we obtain

$$M = n\,(\Delta t) = \frac{2\pi}{365.25}\,(\Delta t)$$

where Δt ≡ the number of days since perihelion passage

and $M = E_s - e_E \sin E_s$

where E_s ≡ eccentric anomaly of the earth about the sun

e_E ≡ eccentricity of the earth's orbit

The true anomaly of the earth in its orbit about the sun is obtained from equation (4) which is

$$\nu = \sin^{-1}\left[\frac{\sqrt{1-e_E^2}\ \sin E_s}{1-e_E \cos E_s}\right]$$

The right ascension of the sun is obtained from

$$\gamma = \tan^{-1}\ [\cos i_S \tan(\nu - V)]$$

where γ ≡ right ascension of the sun

i_S ≡ obliquity of the ecliptic

V ≡ true anomaly of the earth at vernal equinox

The angle between the sun and the orbit plane is the difference of right ascensions which is

$$\beta = [\gamma - \lambda_1]_{MOD\ 2\pi}$$

where λ_1 ≡ right ascension of the node from equation (21)

The rate of change of the angle between the sun and the orbit plane is obtained by differencing the values of β for that day and the preceding day and dividing by the time difference

$$\dot{\beta} = (\beta_i - \beta_{i-1})/(t_i - t_{i-1})$$

where t_i ≡ time of the ascending node for the first rev of the ith day.

For the initial day of the run the value of β_{i-1} will have to be computed for the time, t_{i-1}. The time t_{i-1} will be obtained for the first day by subtracting 24 hours from t_i.

The values of a_1, e_1, and i_1 for the first rev are the same as the input values. This completes the set of orbital parameters which will be considered by the HECTOR routine. The complete set of these parameters is

a_1 ≡ semi major axis

e_1 ≡ eccentricity

i_1 ≡ inclination

Ω_1 ≡ longitude of the ascending node at the time of nodal crossing

ω_1 ≡ argument of perigee at the time of nodal crossing

τ_1 ≡ time of perigee passage

λ_1 ≡ right ascension of the node at the time of nodal crossing

T_s ≡ time of the start of the rev (i.e., the southernmost point on the rev)

β ≡ the angle in the equatorial plane from the ascending node of the vehicle to the sun (orbit plane sun angle)

$\dot{\beta}$ ≡ rate of change of the orbit plane sun angle

3.2.3 Computation of the Elements for Succeeding Revs

When the elements for the jth rev of the day has been computed the elements of the $j + 1^{st}$ rev are obtained from the following:

$$a_{j+1} = a_j \tag{22}$$

$$i_{j+1} = i_j \tag{23}$$

$$e_{j+1} = e_j \tag{24}$$

$$\Omega_{j+1} = [\Omega_j + (\dot{\Omega} - \omega_e) \cdot (2\pi/n)]_{MOD\ 2\pi} \tag{25}$$

$$\omega_{j+1} = [\omega_j + \dot{\omega}\,(2\pi/n)]_{MOD\ 2\pi} \tag{26}$$

$$\tau_{j+1} = \tau_j + 2\pi/n \tag{27}$$

$$T_{s_{j+1}} = T_{s_j} + 2\pi/n \tag{28}$$

It should be noted that T_s is the time of the start of the rev in seconds from midnight and if $T_{s_{j+1}}$ is greater than 86,400 seconds the jth rev was the last rev of the day. The first rev of the next day will be computed in the same manner as just described with the exception of time. For the start of a day, other than the day of launch, the time of perigee passage and the time of the start of the rev will be computed as.

$$\tau_1 = \tau_j + \frac{2\pi + (\dot{\omega} * \Delta t)}{n} - 86,400 \tag{29}$$

$$T_{s_1} = T_{s_j} + 2\pi/n - 86,400 \tag{30}$$

where

τ_j ≡ time of perigee passage of the last rev of the preceeding day

T_{s_j} ≡ time of the start of the last rev of the preceeding day

Δt ≡ $(86,400 - t_o)$ for the second day where t_o is time of injection and

≡ 86,400 for all days greater than the second

3.2.4 Compute Ground Track and Swath Dates for One Rev

The subroutine SWATH will be called to generate a table of visibility swath information for a specified vehicle. The time of perigee passage (TP_{VEH}), mean motion (n_{VEH}) and the eccentric anomaly at the ascending node (E_Ω) will have been computed by the calling routine and available in common. For one rev, the subroutine SWATH will compute vehicle position data at fixed increments of eccentric anomaly. For each vehicle position, the swath data in terms of latitude and longitudinal difference from the node will be computed for a series of cross-track scan angles.

The vehicle position data computations for the ith point in the orbit is initiated by the relationship:

$$E = E_\Omega + \Delta E * (i-1) \tag{31}$$

where,

ΔE = increment for orbit

Then, $T = (E - e \sin E)/n_{VEH} + TP_{VEH}$

where,

e = orbit eccentricity for the vehicle from the ephermeris data.

The sine and cosine of the true anomaly (υ), the argument of latitude (ω_T) and the vehicle altitude (H) can be computed by:

$$\sin(\nu) = \sqrt{1 - e^2} \sin E/(1-e \cos E) \tag{32}$$

$$\cos(\nu) = (\cos E - e)/(1-e \cos E) \tag{33}$$

$$\omega_T = \nu + \omega_P \tag{34}$$

$$H = ER - SM(1-e^2)/(1+e \cos \nu) \tag{35}$$

where

ω_P = argument of perigee for the vehicle from the ephemeris data

SM = semi-major axis for the vehicle from the ephemeris data

ER = radius of the earth in the units of SM

The vehicle geocentric latitude is

$$Z_V = \sin \omega_T * \sin IN \tag{36}$$

$$\lambda_V = ARSIN\ (Z_V) \tag{37}$$

where

IN = orbit inclination for the vehicle from the ephemeris data

The right ascension of Greenwich and the longitude of the node at the time of the ascending node for the point being computed can be obtained from the relationships given by Equation (21).

The cartesian components of the instantaneous node vector in earth centered rotating (ECR) coordinates are:

$$X_N = \cos \phi_\Omega \tag{38}$$

$$Y_N = \sin \phi_\Omega \tag{39}$$

The vehicle position longitude and ECR components can be obtained from:

$$\phi_V = \phi_\Omega + ARTAN\ (\tan \omega_T \cos IN) + K * \pi \qquad \text{(Radians)} \tag{40}$$

$$X_V = \cos \lambda_V \cos \phi_V \tag{41}$$

$$Y_V = \cos \lambda_V \sin \phi_V \tag{42}$$

where $K = 1$ if $90^\circ \leq \omega_T \leq 270^\circ$, otherwise $K = 0$.

The vectors $\overline{N}$ and $\overline{Z}_B$ are defined to be:

$$\overline{Z}_B = (-V_X)i + (-Y_V)j + (-Z_V)k \tag{43}$$

$$\overline{N} = (X_N)i + (Y_N)j \tag{44}$$

where i, j, k are the ECR coordinate axes directions. The unit vector $\overline{Y}_B$ is then defined to be

$$\overline{Y}_B = (\overline{N} \times \overline{Z}_B) / \left| \overline{N} \times \overline{Z}_B \right| \tag{45}$$

The vector $\overline{P}_V$ is defined to be

$$\overline{P}_V = (D * X_V)i + (D * Y_V)j + (D * Z_V)k \tag{46}$$

where $D = 1 + H$ with H in earth radii.

The visibility swath data can be obtained for the vehicle position by computation of the latitude and longitude from the node of the intersection of the earth and a series of rays from the vehicle at various cross-track scan angles. For the jth scan angle ($j = 1, 2, 3$)

$$L_Y = \sin(SA_j) \tag{47}$$

$$L_Z = \cos(SA_j) \tag{48}$$

$$\overline{L} = (A)(M) \tag{49}$$

where A is the two element row matrix (L_Y, L_Z)

M is the 2x3 matrix $\begin{pmatrix} \overline{Y}_B \\ \overline{Z}_B \end{pmatrix}$

$$S = -B - \sqrt{B^2 - H}$$

where $B = \left(\overline{L} \cdot \overline{P}_V\right) / \left|\overline{L}\right|$

and H is in earth radii.

If the value of S is negative, an intersection of the vector at that scan angle does not occur. If S is positive:

$$\overline{R} = \overline{P}_V + \frac{S\,\overline{L}}{\left|\overline{L}\right|} = (R_1)i + (R_2)j + (R_3)k \tag{50}$$

$$\lambda_j = \text{ARSIN}(R_3) \tag{51}$$

$$\phi_j = \text{ATAN 2}(R_2, R_1) - \phi_\Omega \tag{52}$$

$$\tau_j = T - TA \tag{53}$$

These relationships yield the latitude, longitudinal difference and time relative to the node for the scan angles. A table containing these scan angle data along with the vehicle altitude, H, for each vehicle position data point may be output at completion of the processing and is written on the swath file.

3.2.5 Compile Nodal Search Data

The parameters needed for potential access determination in SAGE are:

$$\Omega_{MIN} = MIN\ (\Omega_1,\ \Omega_2) \tag{54}$$

$$\Omega_{MAX} = MAX\ (\Omega_1,\ \Omega_2) \tag{55}$$

where

$$\Omega_1 = 2\pi + \phi_3 - 2\phi_2$$

$$\Omega_2 = 2\pi + \phi_1 - 2\phi_1$$

and

ϕ_j obtained from Equation (52).

The Ω_{MIN}, Ω_{MAX} parameters are a function of latitude and the data represents a conservative estimate for nodal bounds for a potential access condition.

REVTAB receives through common a time ordered table defining the vehicle position and the visibility swath. For each time point the following data is provided in the table

- Time from the ascending node for the ith entry
- Vehicle altitude
- Vehicle latitude
- Longitudinal difference between the longitude of the node at the time of the node and longitude of the vehicle subpoint
- Swath data for each of scan angles
    - Latitude of scan angle vector intersection with the earth
    - Longitudinal difference between the node and the intersection of the scan vector with the earth

REVTAB will reformat the above data to provide the information in terms of latitude rather than time. The initial effort of REVTAB in generating the data for the table is to interpolate and compute the vehicle altitude, vehicle longitudinal difference and the time from the ascending node for each integer degree value of vehicle latitude. Since the vehicle crosses each latitude twice, once ascending and once descending, two sets of data would be needed to define both day and night passage. However, since a sun synchronous orbit is given only descending, passage is processed.

The swath data is processed in the same general manner. For each of the scan angles, the longitudinal difference between time from the ascending node are obtained by interpolation for each integer degree of scan angle vector latitude. The minimum and maximum latitudes for which table values can be obtained are the latitudes equal to the orbit inclination, IN (or 180° - IN if IN > 90°). For latitude values outside the range $\pm$ IN, the table entries will be zero to insure the proper subsequent processing of the data.

4.0 OUTPUT

4.1 GENERAL

There are five media for output. These are:

1. Intermediate debug print for swath calculations
2. Ephemeris display report
3. Echo print of card images
4. Swath table file
5. Swath reference file

All debug output from HECTOR will be deleted.

4.2 INTERMEDIATE DEBUG PRINT

The following debug prints occur if the input flag LIDEBG(1) is set to .TRUE.

1. For each vehicle and day

   SWATH OUTPUT - LAT. AND DELTA LONG FOR 3 POINTS, ALT AND TIME FOR LATITUDE POINT
   XXX.XXXXXXX XXX.XXXXXXX ⟶ for 6 more values XXXX

The following debug print occurs if LIDEBG(4) is .TRUE.

1. For each vehicle and day:

   MXLAT, MNMX, MXMN, NNLAT=XXXX XXXX XXXX XXXX

   For each of 3 scan angles:

   SLAM(361, L)=XX.XXXXXXXX ⟶ 2 more values
   SLAM(362, L)=XX.XXXXXXXX ⟶ 2 more values

2. The following debug print is produced for each lat. above minimum of swath edges at North Pole, for each lat. from lowest swath edge at North Pole to highest swath edge at South Pole and for each lat. from highest swath edge at South Pole to lowest swath edge at South Pole.

   For each vehicle and day and each of 3 scan angles, these data are printed:

   LATITUDE, TIME, and DLONG.

3. For each vehicle and day and for each of 3 scan angles, these data are printed:

   REVTAB OUTPUT - TIME AND DELTA LONG FOR SCAN ANGLES AND ALT

   TIME, DLONG, ALT and LATITUDE

The following debug print of swath file records occurs if L1DEBG(2) is .TRUE. and if the day is within one of the PRTI5 print selection pairs.

1. A dump of each record on the swath reference and swath table is given which meets the above criteria. The records would be printed in pairs as follows:

SWATH REFERENCE RECORD VEHICLE n DAY nnn

WP	NREV	TMNODE	CARUS
±X.XXXXXXE±YY	1	nnnnn	±X.XXXXXXE±YY
	2	↓	↓
	|		
	17		

SWATH TABLE RECORD VEHICLE n DAY nnn

LATNO	IALT	TIME(1)	TIME(2)	TIME(3)	DLONG(1)	DLONG(2)	DLONG(3)
65	nnnn	nnnnn	nnnnn	nnnnn	±X.XXXXXXE±YY →		
64	↓	↓	↓	↓	↓	↓	↓
|							
-65							

4.3 REPORTS

There is only one report which will be printed and only if there are non-zero entries in IPPI5 input data array. The particular days that will be printed are under control of this array.

FORMAT:

EPHEMERIS DISPLAY FOR DAY NNN

VEHICLE N

SEMI-MAJOR AXIS	=	XXXXXX.XXX
ECCENTRICITY	=	XX.XXXXXXXX
PERIOD	=	XXXXXX.XXX
INCLINATION	=	XXXX XX XX
ARGUMENT OF PERIGEE	=	XXXX XX XX
RT. ASC. GREENWICH	=	XXXX XX XX

REV	NODE TIME	NODE LONGITUDE
1)	XXXXXXX.XXXX	XXXX XX XX
2) ↓	↓	↓
17)	XXXXXXX.XXXX	XXXX XX XX

4.4 DATA FILES

There are two output files. These are the swath table and the swath reference. See Section 2.4 of the Users Manual for a description of these files.

Record Length: 80 bytes or 55 words

Block Factor: 4

File Size: 197,820 bytes or 60,445 words

Usage: This file is used in conjunction with the swath table by SAGE to determine when and if a segment is accessed. The swath reference file is searched for a day and rev entry within a supplied delta of a given nodal longitude and time. The day obtained in this manner is used to read the desired record from the swath table. The calculations on the data can then be performed to obtain a further check for access.

4.5 ECHO PRINT OF INPUT CARD IMAGES

The control card input to EPHEM is printed after all nominal values have been stored.

HEADER

AAAA——————————————————>AAA

ICASE	IVEH	NODAY	INLAT	ISLAT	LIDEBG	IPPIS	START/STOP	ISOSTR
NNNN	NN	NNN	NN NN	NN NN	LLLL	NNN_1 NNN_2----	NNN_{10}	NN_{01}

SA			IVITIM			IV2TIM			
NNN.N	NNN.N	NNN.N	NN NN	NN/NN	NN NN	NN NN	NN/NN	NN NN	02

ORBIT 1					ORBIT 2				
NNNN.N	NNNN.N	NNN.	NNN.	NNN.	NNNN.N	NNNN.N	NNN.	NNN.	NNN. 03

5.0 ERROR PROCESSING

5.1 GENERAL

The program will attempt to find as many sources of error during the input card processing as possible. The program w ll continue checking for further input errors upon detecting any input error. Since most of the computations are contained in subroutines which already exist, the philosophy of continuing processing after detection will be retained. There are two levels of error. These are:

Level 1 - continue processing

Level 2 - job fatal

When a level 1 error occurs, the program will print an informative message and continue. If such an error occurs during the calculation phase, a printout of key data quantities is given in addition. When a level 2 error occurs, the program will print an informative message and return control back to the computer system.

Level 2

1. A check is made to see if NODAY is between 1 and 549. Message:

   ***NODAY IS OUT OF RANGE. IT IS NOT BETWEEN 1 AND 549

2. A check is made to see if INLAT and ISLAT are each between 0 and 65. In addition, ISLAT(1) (INLAT(1)) must be ≤ ISLAT(2), (INLAT(2)). Message:

   ***INLAT OR ISLAT IS OUT OF RANGE. THEY ARE NOT BETWEEN 0 AND 65 OR THE FIRST VALUE OF EITHER ONE IS NOT LESS THAN OR = TO THE SECOND.

3. A check is made to make sure the number of latitude points is less than 101 as specified by ISLAT and INLAT. Message:

***THE TOTAL NUMBER OF LATITUDE POINTS SPECIFIED BY ISLAT AND INLAT EXCEEDS 100.

4. A check is made to see if each entry pair for IPPI5 has entries between 1 and 549 and the first entry of a pair is ≤ to the second. Message:

   ***IPPI5(N) AND IPPI5(N+1) ARE OUT OF RANGE. THEY ARE NOT BETWEEN 1 AND 549 OR THE VALUE OF THE FIRST ENTRY IN A PAIR IS NOT LESS THAN OR = TO THE SECOND.

5. IVEH is checked to be between 1 and 2. No more than 2 vehicles can be processed by this program. Message:

   ***IVEH IS NOT BETWEEN 1 AND 2.

6. A check is made to see if each SA entry is in the range of -10 to +10 and is in ascending order and unequal to each other. Message:

   ***EITHER AN SA ENTRY IS NOT BETWEEN 0 AND ABS(10) OR ENTRY(N) IS NOT LESS THAN ENTRY(N+1)

7. A check is made to make sure IV1TIM(1) or IV2TIM(1) is greater than 1963. Message:

   ***EITHER IV1TIM(1) OR IV2TIM(1) IS NOT GREATER THAN 1963.

8. ORBIT1(1) and ORBIT2(1)-a are checked to be in the range of 6650 and 7700 kilometers. Message:

   ***EITHER ORBIT1(1) OR ORBIT2(1) IS NOT BETWEEN 6650 AND 7700 KILOMETERS.

9. A check is made to make certain that ORBIT1(2) and ORBIT2(2)-e is in the range .00001 and .15. Message:

   ***EITHER ORBIT1(2) OR ORBIT2(2) IS NOT BETWEEN .00001 AND .15.

10. A check is made to make sure that ORBIT1(4) and ORBIT2(4)-Ω is in the range 0 and 360. Message:

***EITHER ORBIT1(4) OR ORBIT2(4) IS NOT BETWEEN 0 AND 360.

11. A check is made to make sure that ORBIT1(5) and ORBIT2(5)-ω is in the range 0 and 360. Message:

***EITHER ORBIT1(5) OR ORBIT2(5) IS NOT BETWEEN 0 AND 360.

12. A check is made to make sure that ORBIT1(6) and ORBIT2(6)-τ is in the range ± 65. Message:

***EITHER ORBIT1(6) OR ORBIT2(6) IS NOT BETWEEN -65 AND +65.

13. A check is made to make sure IV1TIM(4) or IV2TIM(4) specifies an hour such that the local vehicle passage time is between 700 to 1700 hours. Message:

***IV1TIM(4) OR IV2TIM(4) SPECIFIES A LOCAL VEHICLE PASSAGE TIME NOT BETWEEN 700 AND 1700 HOURS.

14. A check is made to make sure IV1TIM(3) or IV2TIM(3) is in the range 1 and 31. Message:

***IV1TIM(3) OR IV2TIM(3) IS NOT BETWEEN 1 AND 31.

15. A check is made to make sure IV1TIM(2) or IV2TIM(2) is in the range 1 and 12. Message:

***IV1TIM(2) OR IV2TIM(2) IS NOT BETWEEN 1 AND 12.

16. If one or more of the above errors occur, the following message will print:

***THIS JOB IS ABANDONED DUE TO THE FACT THAT 1 OR MORE FATAL INPUT ERRORS OCCURRED.

5.3 PROCESSING ERRORS

Level 1

1. A check is made in subroutine HECTOR after attempting to solve for the eccentricity anomaly that the iteration loop converged on a solution. Message:

   ***ERROR (KEPLER) SOLUTION FOR ECC. ANOMALY DID NOT CONVERGE AFTER 50 ITERATIONS.

2. A check is made in subroutine REVTAB to determine if an anomaly occurred in calculating crosstrack latitudes. If so, the following message is printed followed by data:

   REVTAB ERROR - ANOMALY IN CROSSTRACK LATITUDES

   Values for K, L, AA, JJ, DEL, XI, INT, NTRY, and LLL are then printed.

3. A check is made in subroutine SWATH to make sure the iteration count of 5 is not exceeded in computing geocentric latitude and delta longitude. Message:

   Iteration limit on SWATH. Values for XLONGN, TEM, DXLON, R(1), R(2), IFLG are then printed.

5.4 INPUT/OUTPUT ERRORS

For sequential I/O the FORTRAN system on the UNIVAC or PDP takes control and prints a message identifying the problem and will either continue processing or abandon the job. If processing continues, the system counts the number of times this error re-occurs and if it happens a certain number of times, the system will abandon the job.

For direct access I/O, the UNIVAC or PDP D.A. I/O package prints out an informative message, sets an error flag and allows processing to continue. In this program the swath table file is the only direct access file. Immediately after the informative message, the following message will print:

***AN IRRECOVERABLE I/O ERROR HAS OCCURRED ON WRITING A RECORD TO THE SWATH TABLE. THE JOB IS BEING ABANDONED.

PART II

COMMON BLOCK DEFINITIONS

COMMON STORAGE ALLOCATION

Name CONS Size _____ Page _____ of _____

Function Contains constants and input data

Name	Dimen-sion	For-mat	Description	Sym-bol	Units
WE	1	R	.72921151E-4		
NVEH	1	I	No. of vehicles 2		
NODAY	1	I	549 no. of days in the run		
IVITIM	6	I	Vehicle 1 injection time no. nominal		
IV2TIM	6	I	Vehicle 2 injection time no. nominal		
ORBIT1	6	R	No nominals orbital elements vehicle 1		
ORBIT2	6	R	No nominals orbital elements vehicle 2		
KO	1	I	Output print unit = 6		
PI	1	R	3.1415926	π	
CJ2	1	R	.10823E-2		
CMU	1	R	.15362204E-5		
PI2	1	R	1.5707963	$\pi/2$	
PI32	1	R	4.712389	$3/2\pi$	
IORBIT	1	I	Orbit option set to 2		
SA	3	R	Scan reference angle no nominal		
LIDEBG	4	L	Debug flags nominal is false		
RADIAN	1	R	57.29578		

COMMON STORAGE ALLOCATION

Name CONS Size _____ Page 2 of _____

Function _____

Name	Dimension	Format	Description	Symbol	Units
ECCE	1	R	.167263E-1		
CIS	1	R	.409280		
CNUV	1	R	1.35695		
RADIUS	1	R	Radius of earth 6376.436		KILOM.
HEADER	20	R	Header for printing initialized to blanks		
ICASE	1	I	Case no. no nominal		
IPPIS	10	I	Report day print range nominal→1, 549, 8*0		
NAME	2	R	'SWATHbTB'		
NAME1	2	R	'SWATHbREF'		
NLINE	1	I	Line count for print set to 99		
NPAGE	1	I	Page no. set to 0		
TWOPI	1	R	6.2831852	2π	
ALPHAM	1	R			
THES	1	R			
INP	1	I	Card input unit = 5		
LINMAX	1	I	Max no. of lines per page set to 39		
ISWTB	1	I	8-swath table file unit		
ISWR	1	I	9-swath ref. file unit		

COMMON STORAGE ALLOCATION

Name FLAG Size _____ Page _____ of _____

Function _____

Name	Dimen-sion	For-mat	Description	Sym-bol	Units
IFIRST	1	I	First time flag, ≠ 0 - first time, = 0 - not first time		
IVEH	1	I	No. of vehicles = 2 max		
ISTART	1	I	Case start date in days since 1950		
IEND	1	I	Case end date in days since 1950		
IDEND	1	I	Current noday for a given vehicle		

COMMON STORAGE ALLOCATION

Name SWATH Size 407 words Page ____ of ____

Function ____

Name	Dimen-sion	For-mat	Description	Sym-bol	Units
INLAT	2	I	North latitude study band definition		DEG
ISLAT	2	I	South latitude study band definition		DEG
ISOSTR	1	I	Δ with respect to north start for south latitude start		MONTH
NLAT	1	I	No. of latitudes (N and S) to be studied		
LATNO	100	I	List of latitudes from North to South for swath		DEG
IALT	100	I	From alt list of corresponding altitudes with LATNO		KILOM * 10
IGRDI	17	I	For FOC ground sta. ind.		
DLONMN	100	I	Minimum Δlongitude table		RAD
DLONMX	100	I	Maximum Δ longitude table		RAD
TIME1	100x3	R	Data from time ready for writing - matches LATNO		
DLONG1	100x3	R	Data from DLONG ready for writing matches LATNO		

COMMON STORAGE ALLOCATION

Name NREV Size _____ Page _____ of _____

Function _____

Name	Dimen-sion	For-mat	Description	Sym-bol	Units
NREV	2	-	Count of rev's for each vehicle		
NREVCT	1x2	I	Equivalent to NREV (1)		
XLONG	1x2	R			

COMMON STORAGE ALLOCATION

Name ACCEPH Size 47 words Page ____ of ____

Function ____

Name	Dimension	Format	Description	Symbol	Units
TMNODE	17x2	R	Time of crossing for ascending node		SEC
SM	1x2	R			
EC	1, 2	R			
XIN	1, 2	R			
RTASC	1, 2	R			
WP	1, 2	R			
TNDE	1, 2	R			
RG	1	R			

COMMON STORAGE ALLOCATION

Name ACCDAT Size 2898 words Page ____ of ____

Function Contains swath data

Name	Dimen-sion	For-mat	Description	Sym-bol	Units
TV	362	R			
HV	362	R			
SLAM	362x3	R			
SPHI	362x3	R			
ENP	2	R			

COMMON STORAGE ALLOCATION

Name REVSTH Size 2534 words Page ____ of ____

Function ____

Name	Dimen-sion	For-mat	Description	Sym-bol	Units
TIME	181x3x2	R			
DLONG	181x3x2	R			
ALT	181x2	R			

PART III

LIST OF SUBROUTINES

AND

SUBROUTINE CALL STRUCTURE

LIST OF ROUTINES IN EPHEM

NAME	FUNCTION
1. DAY	Computes IDAY - DAY no. of the year
2. PIMOD	Converts an angle in radians to be between 0 - 2π
3. LFPA	Computes right ascension and ZULU day
4. CLDAY	Computes month and day of current year
5. PAGER	Restores pages and prints header
6. KEPLER	Computes eccentric anomaly
7. ACCGEN	Controls calculation of the orbital swath
8. HECTOR	Compute ephemeris data
9. REVTAB	Generates swath data vs latitude points
10. SWATH	Computes swath table
11. ALPHA	Computes sun angle
12. DEGMOD	Converts an angle in radians to degrees
13. INPROC	Reads and checks input data control cards
14. LATSEL	Forms the latitude band for swath table output
15. WRITSW	Writes a record on the swath table and swath reference file.
16. PRTI5	Print the ephemeris report
17. MAIN	Main control routine for the program
18. ERRDET	Process's errors and prints messages

EPHEM SUBROUTINE CALL STRUCTURE

```
MAIN
    INPROC
        EJECT
        ERRDET
            PAGER
    LFPA
    DAY
    PAGER
    HECTOR
        CLDAY
        LFPA
        PAGER
        KEPLER
        ALPHA
        PIMO0
    WRITSW
    ACCGEN
        SWATH
            PIMOD
            PAGER
            EJECT
        REVTAB
            PAGER
        LATSEL
            PIMOD
        WRITSW
    PRTI5  PRTI5
        DEGMOD
        EJECT
        PAGER
    EJECT
```

PART IV

SUBROUTINE DESCRIPTIONS

SUBROUTINE KEPLER

Purpose:

This routine solves KEPLERS equations for eccentric anomaly (XE), given eccentricity (XECC) and mean anomaly (XM). The equations for the elliptic case is XM = -XECC* & IN(XE).

Calling Sequence:

Call KEPLER (XM, XECC, XE, ERROR)

SUBROUTINE ACCGEN

Purpose:

This routine initializes data for access's, computes preliminary orbit parameters, computes the eccentric anomally at ascending node and time of perigee passage. It also controls the calculation of the orbital swath and latitude interpolation tables.

SUBROUTINE HECTOR

Purpose:

This routine generates the ephemeris data for two vehicles for one day. The ephemeris is maintained as a set of orbital elements.

SUBROUTINE REVTAB

Purpose:

This routine receives as input the swath data table from the subroutine swath. REVTAB reformats the table to obtain the data at each integer value of latitude.

SUBROUTINE SWATH

Purpose:

This routine utilizes the ephemeris data for the day specified to generate the latitude and delta longitude data for the scan angles for one rev. The data is generated at one degree increments of eccentric anomaly, beginning at the ascending node.

Subroutine INPROC

ENTRY POINT: Call INPROC

PURPOSE: This routine reads the job header card, the program control card and 2 data cards as discussed in the problem description document. It also calls ERRDET to find input errors then performs initial calculations.

INPUT: All data in the input data cards as discussed in Section 2.0 - Problem Description.

OUTPUT: CONS COMMON

ICASE, NODAY, NVEH, HEADER, LIDEBG, IV1TIM, IV2TIM, ORBIT1, ORBIT2, SA, INP

FLAG COMMON

IVEH

SWAT COMMON

INLAT, ISLAT, IPPIS, ISOSTR, LATNO

SUBROUTINES USED:

CALL ERRDET

METHOD/PROCEDURE

- Read NODAY, IVEH, INLAT, ISLAT and IPPI5 into a temporary storage area and the rest of the data directly into working storage. Move the above items into working storage only if $\neq 0$. Call ERRDET to check input for errors. Perform initial calculations per Sec. 3.4.3. and in last paragraph item 3.

Subroutine LATSEL

ENTRY POINT: Call LATSEL

PURPOSE: To move swath data from a full 180° swath table storage to an area for writing on the swath table. This routine also computes the minimum and maximum Δ longitude for all required latitudes.

INPUT: SWAT COMMON

NLAT, LATNO, DLONMX, DLONMN

REVSTH COMMON

TIME, DLONG, ALT

CONS COMMON

TWOPI

OUTPUT: SWAT COMMON

IALT, DLONMN, DLONMX, TIME1, DLONG1

SUBROUTINES USED: None

METHOD/PROCEDURE

See the 4th and 5th paragraphs of Section 3.4 item 4.

Subroutine WRITSW

ENTRY POINT: Call WRITSW (IOPT)

PURPOSE: To initialize, write and terminate output records on the swath table file - ISWTB and swath reference file - ISWR. A record is written for each vehicle and file.

INPUT: CONS COMMON

NAME, NAME1, ICASE, NVEH, NODAY, IV1TIME, IV2TIM, ISWTB, ISWR

CALLING SEQUENCE: IOPT

HECCOM COMMON

XOMEGD(2) - 21th LOC, XWD - 19th LOC

ACCNUM COMMON

KDAY, KVEH

ACCEPH COMMON

WP (1, KVEH), TMNODE (i, KVEH), i = 1, 17

NREV COMMON

NREV (1, KVEH)

ICARUS COMMON

(CARUS (i, 1, j), i = 1, 3, j = 1, 2), (CARUS (4, i, KEVH), i = 1 - 17)

SWAT COMMON

INLAT, ISLAT, ISOSTR, (LATNO (i), IALT(i), TIME1 (i, n), DLONG1(i, n), n = 1, 3), DLONMX(i), DLONMN(i), i = 1, NLAT)

OUTPUT: See swath table and swath reference table output formats in Problem Description document

DACDC COMMON: INDEXR

SUBROUTINES USED:

FOR CDC - OPENMS, READMS, WRITMS, SLOSEMS

D. A. PACKAGE

FOR UNIVAC - The define file PACKAGE

METHOD/PROCEDURE

The routine functions according to the value in IOPT

For IOPT = 1 - initialization

SWATH TABLE

a. A define file is given to initialize pointers and open the file.

b. Data is moved from common storage to the output array A(800).

c. The header record is written

SWATH REFERENCE - Rec. No. 1

a. A rewind is given.

b. Header record written.

For IOPT = 2 - write detail records

SWATH TABLE

a. Data for detail record is moved into "A"

b. Write detail record, get rec. no. from formula in Problem Definition.

SWATH REFERENCE

a. Write detail record

For IOPT = 3 - write swath table trailer record

a. Data for trailer record is moved into "A"

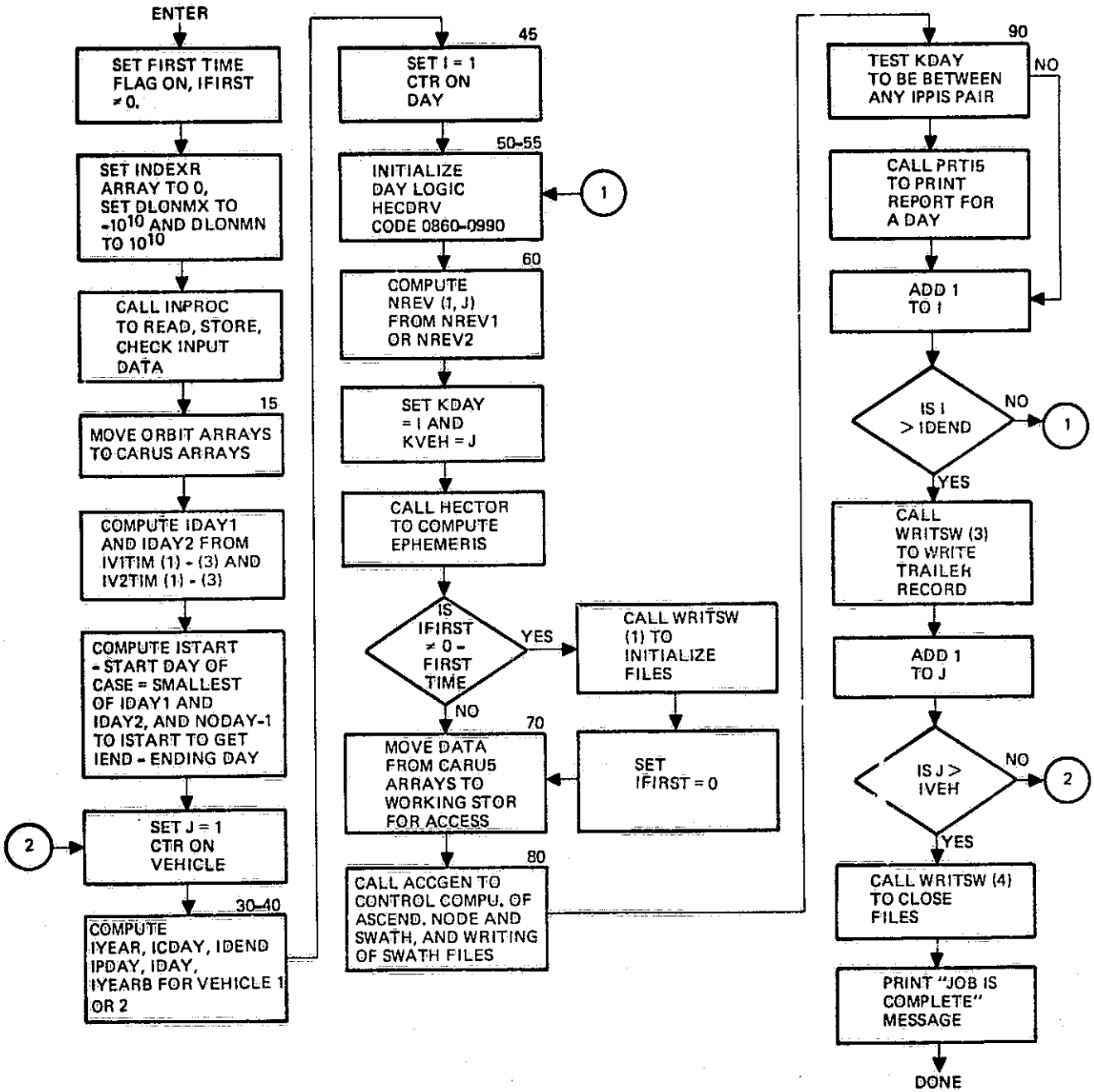
b. Write the record, rec. no. is obtained from formula in Problem Definition

For IOPT = 4 - close files

SWATH TABLE - Automatic close

SWATH REFERENCE - Write EOF and rewind.

SUBROUTINE MAIN BLOCK DIAGRAM



Subroutine ERRDET

ENTRY POINT: Call ERRDET

PURPOSE: This routine performs all input error checking for the data on cards. It will continue to check for all errors even though 1 or more errors were detected. Having found at least one error the program will stop.

INPUT: CONS COMMON

NODAY, SA, IV1TIM, ORBIT1, ORBIT2

SWAT COMMON

INLAT, ISLAT, IPPIS, S

FLAG COMMON

IVEH

OUTPUT: Error messages as shown in Section 5.2

METHOD/PROCEDURE

When an error occurs a counter is updated. When the last error has been checked this counter is tested if = 0 then a normal return is executed, otherwise a termination message is printed and a stop is executed. The errors to be checked are found in Section 5.2.

SUPPLIED UTILITY ROUTINES

Routine Day

Call Day (IYMD, IDAY)

Given IYMD (3) where

IYMD (1)	IS Day No.
IYMD (2)	IS Month No.
IYMD (3)	IS Year No.

Compute year day no. in IDAY

Routine PIMOD

Call PIMOD (A)

Convert ±A in radians to an angle 0-2π

Routine SOL (Entry ALPHA)

Call ALPHA (IFLAG)

For emphemeris usage as called by hector

computes ALPHAM and ALPHAT and IFLAG = 1

Routine PAGER (Entry Eject)

Call PAGER (NLINES)

Updates line count in NLINE with NLINES

NPAGE = 0 causes page to be restored prior to print.

NPAGE - page no.

HEADER- 80 char. 20A5

ICASE- case no.

KO - 6 print unit

INMAX is max no. of lines allowed

Initially NLINE should be set > LINMAX and NPAGE = 0

SUPPLIED UTILITY ROUTINES
(CONTINUED)

Call EJECT (NLINES)

Causes page to be restored automatically and then prints headers.

Routine CLDAY

Call CLDAY

Given IDAY-DAY no. of the year compute in LMO-the month and in LDA the day no.

Need: IYEAR = 0 - Leap Year, ≠ 0 not Leap Year

Routine KEPLER

Call KEPLER (XM, XECC, XE, ERROR)

Given XM - Mean anomaly, XECC - eccentricity

Compute: E-eccentric anomaly, error = 0 means OK

Routine LFPA

Call LFPA [FLDA, LMO, LYR, ALFGM (can be dummy), DAYS]

Given: FLDA - day of month no., LMO - month no.,

LYR - year no. compute ALFGM - right ascension and

DAYS - Zulu day no.

Routine DEGMOD

Call DEGMOND (RAD, IDEG)

Given: angle rad in radians store the angle in deg., min., sec., in IDEG(1) - (3).

Routine FZULU

Call FZULU (IOATE, IOUT)

Given Zulu date in IDATE, compute year, month and day in IOUT(1) - IOUT(3).

Routine RDMIA

Call RDMIA(FL, U)

Given double precision random no. seed in FL, comoute random no. U (0-1) based on uniform distribution.

PART V

SUBROUTINE LISTINGS

```
      SUBROUTINE ACCGEN                                                 ACCGEN
C                                                                       ACCGEN
C                                                                       ACCGEN
      COMMON /CONS/                                                     CONS
     1ECCF,CIS,CNUV,RADIUS,HEADER(20),NAME(2),NAME1(2),TWOPI,ALPHAM,    CONS
     2THES,VF,ORBTT1(6),ORBTT2(6),PI,CJ2,CMU,PI2,PI32,SA(3),RADTAN,     CONS
     3ICASE,IPPIS(10),NLINE,NPAGE,INP,LINMAX,ISWTR,ISNR,NVEH,NUDAY,     CONS
     4IV1TIM(6),IV2TIM(6),KD,IORBTT,LIDEBG(4)                           CONS
      REAL NAME,NAME1                                                   CONS
      LOGICAL LIDEBG                                                    CONS
C                                                                       ACCGEN
      DIMENSION MAXLT(2),MINLT(2)                                       ACCGEN
      DIMENSION DELNDE(2),PER(2)                                        ACCGEN
      DIMENSION TIMNDE(19,2),XLONND(19,2),MXREV(2)                      ACCGEN
      COMMON/ACCNUM/ACCNUM(100)                                         ACCGEN
      EQUIVALENCE (ACCNUM(   1),KVEH)                                   ACCGEN
     X,           (ACCNUM(   3),NDSCN)                                  ACCGEN
     X,           (ACCNUM(   4),NADSCN)                                 ACCGEN
     X,           (ACCNUM(   6),XN )                                    ACCGEN
     X,           (ACCNUM(   7),PER(1))                                 ACCGEN
     X,           (ACCNUM(   9),TIMNDE(1,1))                            ACCGEN
      EQUIVALENCE (ACCNUM(  47),XLONND(1,1),ARRAY1(1))                  ACCGEN
     X,           (ACCNUM(  66),ARRAY2(1))                              ACCGEN
     X,           (ACCNUM(  85),DELNDE(1))                              ACCGEN
     X,           (ACCNUM(  87),MXREV(1))                               ACCGEN
     X,           (ACCNUM(  89),KDAY)                                   ACCGEN
     X,           (ACCNUM(  90),E )                                     ACCGEN
     X,           (ACCNUM(  91),TPER)                                   ACCGEN
     X,           (ACCNUM( 92),MAXLT(1))                                ACCGEN
     X,           (ACCNUM( 94),MINLT(1))                                ACCGEN
     X,           (ACCNUM(  96),IADF)                                   ACCGEN
C                                                                       ACCGEN
C                                                                       ACCGEN
      COMMON /NPEV/                                                     NPEV
     1XLONG(1,2),NREV(2)                                                NPEV
      DIMENSION NPEVLT(1,2)                                             NPEV
      EQUIVALENCE (NREV(1),NREVC(1,1))                                  NPEV
      COMMON /ACCEPH/                                                   ACCEPH
     1TIMNDE(17,2),SM(1,2),FC(1,2),XIN(1,2),RTASC(1,2),WP(1,2),TNDE(1,2)ACCEPH
     2,PG(1)                                                            ACCEPH
      COMMON /REVSTH/                                                   REVSTH
     1TIME(181,3,2),DLONG(181,3,2),ALT(181,2)                           REVSTH
      COMMON /FLAG/                                                     FLAG
     1IFIRST,IVEH,ISTART,IEND,IDEND                                     FLAG
      COMMON /SAAT/                                                     SAAT
     1DLONGM(100),DLONMX(100),TIME1(100,3),DLONG1(100,3),TNLAT(2),      SAAT
     2ISLAT(2),ISCSTR,NLAT,LATNO(100),IALT(100),ICHDI(17)               SAAT
C                                                                       ACCGEN
C  INITIALIZE DATA FOR ACCESSES                                         ACCGEN
C                                                                       ACCGEN
C                                                                       ACCGEN
      DO 8 I = 1,3                                                      ACCGEN
      IF ( SA(I) .EQ. 0.0) GO TO 8                                      ACCGEN
    5 NDSCN = I                                                         ACCGEN
    8 CONTINUE                                                          ACCGEN
      NADSCN = NDSCN/2 + 1                                              ACCGEN
      MULTIF = 0                                                        ACCGEN
      I = 1                                                                 *NEW
```

ORIGINAL PAGE IS OF POOR QUALITY

```
C                                                                       ACCGEN  **-1
      XLONG(I+1) = 1000000.                                             ACCGEN
      XLONG(I+2) = 1000000.                                             ACCGEN
C                                                                       ACCGEN
C  PROCESS THE ACCESS FOR EACH VEHICLE                                  ACCGEN
C                                                                       ACCGEN
C                                                                       ACCGEN
C  COMPUTE PRELIMINARY ORBIT PARAMETERS                                 ACCGEN
      I= 1                                                              SET1
      J = KVEH                                                          SET1
C                                                                       ACCGEN
      XN = SQRT(CMU/SM(I,J)**3)                                         ACCGEN
      XECC = EC(I,J)                                                    ACCGEN
      PER(J) = TWOPI/XN                                                 ACCGEN
C                                                                       ACCGEN
C  COMPUTE ECCENTRIC ANOMALY AT ASCENDING NODE                          ACCGEN
C                                                                       ACCGEN
      TRM1 = (1.-XECC**2)/(1.+XECC*COS(-WP(I,J)))                       ACCGEN
      SINE = TRM1*SIN(-WP(I,J))/SQRT(1.-XECC**2)                        ACCGEN
      COSE = (1.-TRM1)/XECC                                             ACCGEN
      E =  ACOS(COSE)                                                   ACCGEN
      IF ( SINE .LT. 0.0) E = TWOPI - E                                 ACCGEN
C                                                                       ACCGEN
C  COMPUTE TIME OF PERIGEE PASSAGE                                      ACCGEN
C                                                                       ACCGEN
      ENDE = E - TWOPI                                                  ACCGEN
      TRM2 = (ENDE - XECC*SIN(ENDE))/XN                                 ACCGEN
      TPER = TNDE(I,J) - TRM2                                           ACCGEN
      TEMP = TNDE(I,J) + PER(J)                                         ACCGEN
 15   IF( TPER .GE. TNDE(I,J) .AND. TPER .LE. TEMP) GO TO 80            ACCGEN
      IF ( TPER .GE. TNDE(I,J) ) GO TO 20                               ACCGEN
      TPER = TPER + PER(J)                                              ACCGEN
      GO TO 15                                                          ACCGEN
 20   TPER = TPER - PER(J)                                              ACCGEN
      GO TO 15                                                          ACCGEN
C                                                                       ACCGEN
C  GENERATE SWATH BOUNDARIES AND BUILD LATITUDE INTERPOLATION TABLE     ACCGEN
C                                                                       ACCGEN
C                                                                       ACCGEN
 80   CONTINUE                                                          ACCGEN
      CALL SWATH                                                        ACCGEN
      CALL REVTAB                                                       ACCGEN
      CALL LATSEL                                                       ACCGEN
      CALL WRITSW(2)                                                    ACCGEN
C                                                                       ACCGEN
C                                                                       ACCGEN
 100  CONTINUE                                                          ACCGEN
C                                                                       ACCGEN
      RETURN                                                            ACCGEN
C                                                                       ACCGEN
C                                                                       ACCGEN
      END                                                               ACCGEN
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
* F-1 ALPHA,1,750925, 58800     , 1.

      SUBROUTINE ALPHA(IFLAG)                                           ALPHA
C                                                                       ALPHA
C                                                                       ALPHA
C  SUBROUTINE SOL PROVIDES THE SUN ELEVATION ANGLE OF AZIMUTH WHEN CALLE ALPHA   **-4
C  FOR SPECIFIC TARGET                                                  ALPHA
C                                                                       ALPHA
C                                                                       ALPHA
C        CALLING SEQUENCES...                                           ALPHA
C                 CALL SOL(PHIT,THET,TM,ELEVS,AZIMS,IFLAG)              ALPHA
C                                                                       ALPHA
C                          INPUT --  PHIT -- LONGITUDE OF TARGET        ALPHA
C                                    THET -- LATITUDE OF TARGET         ALPHA
C                                    TM   -- TIME                       ALPHA
C                                    IFLAG-- =0, COMPUTE SUN AZ/EL      ALPHA
C                          OUTPUT -- ELEVS-- SUN ELEVATION ANGLE        ALPHA
C                                    AZIMS-- SUN AZIMUTH                ALPHA
C                 CALL ALPHA(IFLAG)                                     ALPHA
C                                                                       ALPHA
C                          INPUT --  IFLAG-- =1 COMPUTE RT. ASC. OF TALPHA
C                                                                       ALPHA
C                                                                       ALPHA
      COMMON /CONS/                                                     CONS
     1ECCF,CIS,CNUV,RADIUS,HEADEN(20),NAME(2),NAMEI(2),TWOPI,ALPHAM,    CONS
     2THES,WE,ORBITI(6),ORBIT2(6),PI,CJ2,CMU,PI2,PI32,SA(3),RADIAN,     CONS
     3ICASE,IPPIS(10),NLINE,NPAGE,INP,LINMAX,ISHIR,ISKR,NVEH,NUDAY,     CONS
     4IV1TIM(6),IV2TIM(6),KO,IORBIT,LIDEBG(4)                           CONS
      REAL NAME,NAMEI                                                   CONS
      LOGICAL LIDEBG                                                    CONS
C                                                                       CONS
      COMMON /FLAG/                                                     ALPHA
     1IFIRST,IVEH,ISTART,IEND,IDEND                                     FLAG
      COMMON /HECCOM/                                                   FLAG
     1ALPHAT,XDELTAD(2),XDELTA(2,2),XWD(2),XOMEGD(2),TASCNP(2),XIO(2),  HECCOM
     2PERIOD(2),LMU,LDA,NREV1,NREV2,IYR,IYRB,IDAY,IPDAY,ICDAY,NREVIP,   HECCOM
     3NREV2P,IV1DAY,IV2DAY,IYR                                          HECCOM
C                                                                       SET1
C                                                                       ALPHA
C  COMPUTE DAY OF YEAR (SINCE NOT AVAILABLE IN PREDICTOR)               ALPHA
C                                                                       ALPHA
C  ERROR TERM DUE TO VARIABLE SPEED OF ROTATION OF EARTH                ALPHA
      EP = 0.                                                           ALPHA
C  CALCULATE MEAN ANOMALY OF EARTH                                      ALPHA
      DAY = IDAY                                                        ALPHA
      EM = (DAY-2.)*TWOPI/365.25                                        ALPHA
C  CALCULATE ECC. ANOMALY OF EARTH                                      ALPHA
      EP = .001/RADIAN                                                  ALPHA
      E1 = EM                                                           ALPHA
   10 E2 = E1-((E1-EM-ECCF*SIN(E1))/(1.-ECCF*COS(E1)))                  ALPHA
      ED = ABS(E2-E1)                                                   ALPHA
      E1 = E2                                                           ALPHA
      IF(ED-EP) 20,20,10                                                ALPHA
   20 IF(E1-TWOPI) 40,50,30                                             ALPHA
   30 E1 = E1-TWOPI                                                     ALPHA
   40 IF(ABS(E1-PI)-EP) 41,41,45                                        ALPHA
   45 IF(E1) 46,55,60                                                   ALPHA
   46 E1 = E1+TWOPI                                                     ALPHA
      GO TO 40                                                          ALPHA
C  TRUE ANOMALY OF EARTH                                                ALPHA
```

```
000059          41 XNUF = PI                                                ALPHA
000060             GO TO 70                                                 ALPHA
000061          50 E1 = 0.                                                  ALPHA
000062          55 XNUF = 0.                                                ALPHA
000063             GO TO 70                                                 ALPHA
000064          60 XNUF = 2.*ATAN(SQRT((1.+ECCE)/(1.-ECCE))*TAN(E1/2.))     ALPHA
000065             IF(XNUF) 65,70,70                                        ALPHA
000066          65 XNUF = TWOPI+XNUF                                        ALPHA
000067       C  DECLINATION OF SUN                                          ALPHA
000068          70 THES = ASIN(SIN(CTS)*SIN(XNUF-CNUV))                     ALPHA
000069             DIF = XNUF-CNUV                                          ALPHA
000070             COSDIF = COS(DIF)                                        ALPHA
000071             SINDIF = SIN(DIF)                                        ALPHA
000072             IF(COSDIF) 90,75,90                                      ALPHA
000073          75 IF(SINDIF) 85,80,80                                      ALPHA
000074       C  RT.ASC. OF TRUE SUN                                         ALPHA
000075          80 ALPHAT = PI2                                             ALPHA
000076             GO TO 150                                                ALPHA
000077          85 ALPHAT = 3.*PI/2.                                        ALPHA
000078             GO TO 150                                                ALPHA
000079          90 ALPHAT = ATAN(COS(CTS)*SINDIF/COSDIF)                    ALPHA
000080             IF(COSDIF) 105,95,95                                     ALPHA
000081          95 IF(ALPHAT) 100,150,150                                   ALPHA
000082         100 ALPHAT = TWOPI+ALPHAT                                    ALPHA
000083             GO TO 150                                                ALPHA
000084         105 ALPHAT = ALPHAT+PI                                       ALPHA
000085       C  RT. ASC. OF MEAN SUN                                        ALPHA
000086         150 ALPHAM = (279.69668+360./365.25*DAY)/RADIAN              ALPHA
000087       C                                                              ALPHA
000088        1000 RETURN                                                   ALPHA
000089             END                                                      ALPHA
```

ORIGINAL PAGE IS OF POOR QUALITY

```
@ ELT BLKDAT,1,750922, 69808   , 1

      BLOCK DATA                                                        BLOCK
      COMMON /SLAT/                                                     SWAT
     1DLONMN(100),DLONMX(100),TIME1(100,3),DLONG1(100,3),INLAT(2),      SWAT
     2ISLAT(2),ISOSTR,NLAT,LATNO(100),IALT(100),IGRDI(17)               SWAT
C     CONTAINS INITIAL VALUES FOR COMMON BLOCK QUANTITIES               BLOCK
      COMMON /CONS/                                                     CONS
     1ECCE,CIS,CNUV,RADIUS,HEADER(20),NAME(2),NAME1(2),TWOPI,ALPHAM,    CONS
     2TBLS,WE,ORBIT1(6),ORBIT2(6),PI,CJ2,CMU,PI2,PI32,SA(3),RADIAN,     CONS
     3ICASE,IPPIS(10),NLINE,NPAGE,INP,LINMAX,ISWTB,ISWR,NVEH,NODAY,     CONS
     4IV1TIM(6),IV2TIM(6),KO,TORBIT,LIDEBG(4)                           CONS
      REAL NAME,NAME1                                                   CONS
      LOGICAL LIDEBG                                                    CONS
      COMMON /FLAG/                                                     FLAG
     1IFIRST,IVEH,ISTART,IEND,IDEND                                     FLAG
      DATA IVEH/2/                                                      BLOCK
      DATA WE/.72921151E-4/,NVEH/2/,NODAY/549/,IV1TIM/6*0./,IV2TIM/6*0/,BLOCK
     1ORBIT1/6*0./,ORBIT2/6*0./,KO/6/,PI/3.1415926/,CJ2/.10826E-2/      BLOCK
      DATA                                                              BLOCK
     1CMU/.15302204E-5/,PI2/1.5707963/,PI32/4.712389/,IORBIT/2/,SA/3*0./BLOCK
     2,LIDEBG/4*.FALSE./,RADIAN/57.29578/                               BLOCK
      DATA ECCE/.167263E-1/,CIS/.409280/,CNUV/1.35695/,RADIUS/63/6.436/,BLOCK
     1HEADER/20*4H    /,IPPIS(1)/1/,IPPIS(2)/549/,IPPIS(3)/0/,         BLOCK
     2IPPIS(4)/70/                                                      BLOCK
      DATA NAME/4HSWAT,4HH TB/,NAME1/4HSWA1,4HH RF/,NLINE/99/,NPAGE/0/, BLOCK
     1TWOPI/6.283185Z/,INP/5/,LINMAX/39/,ISWTB/8/,ISWR/9/               BLOCK
      DATA INLAT/5,65/,ISLAT/15,45/                                     SETT
      END                                                               BLOCK
```

```
* FLT CLDAY,1,751013, 57506     , 1

000001          SUBROUTINE CLDAY                                          CLDAY
000002    C                                                               CLDAY
000003    C MODULE - PPCLDAY                                              CLDAY
000004    C OCTOBER 1, 1973                                               CLDAY
000005    C                                                               CLDAY
000006    C                                                               CLDAY
000007    C  JUNE 26,1973                                                 CLDAY
000008    C                                                               CLDAY
000009    C  MODULE PPCLDAY - POINT TARGET PERFORMANCE PREDICTOR          CLDAY
000010    C                                                               CLDAY
000011    C  GIVEN DAY OF YEAR(1-366), CLDAY COMPUTES THE MONTH AND DAY OF MONTH  CLDAY
000012    C                                                               CLDAY
000013          COMMON /HECCOM/                                           HECCOM
000014         1ALPHAT,XBETAD(2),XBETA(2,2),XKD(2),XOMEGD(2),TASCNP(2),XTO(2),  HECCOM
000015         2PERIOD(2),LMO,LDA,NREV1,NREV2,IYR,IYRB,IDAY,IPDAY,ICDAY,NREV1P,  HECCOM
000016         3NRLV2P,IV1DAY,IV2DAY,LYR                                  SETI
000017          EQUIVALENCE (IYR,IYEAR)                                   CLDAY
000018          DIMENSION ITAB(13),ITAB1(13)                              CLDAY
000019          DATA ITAB/1,32,60,91,121,152,182,213,244,274,305,335,     CLDAY
000020         *                 366/                                    CLDAY
000021          DATA ITAB1/1,32,61,92,122,153,183,214,245,275,306,336,    CLDAY
000022         *                 367/                                    CLDAY
000023    C  CHECK FOR LEAP YEAR                                          CLDAY
000024    C                                                               CLDAY
000025          IF(IYEAR.EQ.0) GO TO 20                                   CLDAY
000026    C                                                               CLDAY
000027          DO 10 K=1,12                                              CLDAY
000028          IF(ITAB(K).LE.IDAY .AND. IDAY.LT.ITAB(K+1)) GO TO 40      CLDAY
000029       10 CONTINUE                                                  CLDAY
000030          GO TO 40                                                  CLDAY
000031    C                                                               CLDAY
000032    C  LEAP YEAR                                                    CLDAY
000033    C                                                               CLDAY
000034       20 DO 30 K=1,12                                              CLDAY
000035          IF(ITAB1(K) .LE. IDAY  .AND.  IDAY .LT. ITAB1(K+1)) GO TO 35
000036       30 CONTINUE                                                  CLDAY
000037       35 LMO = K
000038          LDA = IDAY - ITAB1(K) + 1                                 *NEW
000039          GO TO 50                                                  **-1
000040       40 LMO=K                                                     CLDAY
000041          LDA=IDAY-ITAB(K)+1                                        CLDAY
000042       50 RETURN
000043          END                                                       CLDAY
```

```
• ECT DAY,1,750925, 58783        , 1

000001          SUBROUTINE DAY(IYMD,IDAY)                                  DAY
000002     C                                                               DAY
000003     C                                                               DAY      **-7
000004     C  CONVERTS MONTH, DAY TO DAY NUMBER                            DAY
000005     C                                                               DAY
000006          DIMENSION IYMD(3),IMONTH(12),IMON(12)                      DAY
000007     C                                                               DAY
000008          DATA IMONTH/0,31,59,90,120,151,181,212,243,273,304,334/    DAY
000009          DATA IMON/0,31,60,91,121,152,182,213,244,274,305,335/      DAY
000010     C                                                               DAY
000011          MONTH=IYMD(2)                                              DAY
000012          IDAY=IYMD(3)                                               DAY
000013          IYEAR=MOD(IYMD(1),4)                                       DAY
000014          IF(IYEAR .EQ. 0)GO TO 10                                   DAY
000015     C                                                               DAY
000016     C                                                               DAY
000017     C                                                               DAY
000018          IF (MONTH.NE.0) IDAY=IDAY+IMONTH(MONTH)                    DAY
000019     C                                                               DAY
000020        5 RETURN                                                     DAY
000021       10 IF(MONTH .NE. 0) IDAY = IDAY + IMON(MONTH)                 DAY
000022          GO TO 5                                                    DAY
000023     C                                                               DAY
000024          END                                                        DAY
```

```
* FLT DEGMOD,1,750925, 58801    , 1

000001          SUBROUTINE DEGMOD(RAD,IDEG)                                  DEGMOD
000002    C                                                                  DEGMOD
000003    C                                                                  DEGMOD  **-6
000004    C     DEGMOD CONVERTS RADIANS INTO DEGREES, MINUTES, AND SECONDS   DEGMOD
000005    C            AND MODS  THE DEGREES BETWEEN -360 TO +360            DEGMOD
000006    C                                                                  DEGMOD
000007    C     IDEG(1)=DEGREES                                              DEGMOD
000008    C     IDEG(2)=MINUTES                                              DEGMOD
000009    C     IDEG(3)=SECONDS                                              DEGMOD
000010    C     RAD    = RADIANS                                             DEGMOD
000011    C     RADIAN = DEGREES PER RADIAN                                  DEGMOD
000012    C     RDUM   = R*4 TEMPORARY STORAGE                               DEGMOD
000013    C                                                                  DEGMOD
000014          INTEGER IDEG(3)                                              DEGMOD
000015    C                                                                  DEGMOD
000016          COMMON /CONS/                                                CONS
000017         1ECCE,CIS,CNUV,RADIUS,HEADER(20),NAME(2),NAME1(2),TWOPI,ALPHAM, CONS
000018         2THLS,WE,ORBTI1(6),ORBTI2(6),PI,CJ2,CMU,PI2,PI32,SA(3),RADIAN, CONS
000019         3ICASL,IPPTS(10),NLINE,NPAGE,INP,LINMAX,ISWTR,ISWR,NVEH,NODAY, CONS
000020         4IVITIM(6),IV2TIM(6),KO,IORBTT,LIDEBG(4)                      CONS
000021          REAL NAME,NAME1                                              CONS
000022          LOGICAL LIDEBG                                               CONS
000023    C                                                                  DEGMOD
000024    C     CONVERT RADIANS TO DEGREES, MINUTES, AND SECONDS             DEGMOD
000025    C                                                                  DEGMOD
000026          RDUM = RAD*RADIAN                                            DEGMOD
000027          IDEG(1)=RDUM                                                 DEGMOD
000028          RDUM=ABS(RDUM-IDEG(1))*60.                                   DEGMOD
000029          IDEG(2)=RDUM                                                 DEGMOD
000030          IDEG(3)=(RDUM-IDEG(2))*60.+.5                                DEGMOD
000031    C                                                                  DEGMOD
000032          IF(IDEG(3).LT.60) GO TO 10                                   DEGMOD
000033          IDEG(3)=IDEG(3)-60                                           DEGMOD
000034          IDEG(2)=IDEG(2)+1                                            DEGMOD
000035    C                                                                  DEGMOD
000036       10 IF(IDEG(2).LT.60) GO TO 20                                   DEGMOD
000037          IDEG(2)=IDEG(2)-60                                           DEGMOD
000038    C                                                                  DEGMOD
000039          ISIGN=1                                                      DEGMOD
000040          IF(IDEG(1).LT.0) ISIGN=-1                                    DEGMOD
000041          IDEG(1)=IDEG(1)*ISIGN                                        DEGMOD
000042    C                                                                  DEGMOD
000043       20 CONTINUE                                                     DEGMOD
000044    C                                                                  DEGMOD
000045          IDEG(1)=MOD(IDEG(1),360)                                     DEGMOD
000046          RETURN                                                       DEGMOD
000047          END                                                          DEGMOD
```

```
* ELT EJECT,1,760331, 59923     , 1

000001          SUBROUTINE EJECT(NLINES)                                EJECT
000002     C    RESTORE PAGE AND PRINT PAGE HEADER                      EJECT
000003          COMMON /CONS/                                           CONS
000004         1ECCE,CIS,CMUV,RADIUS,HEADER(20),NAME(2),NAME1(2),TWOPI,ALPHAM,   CONS
000005         2THLS,ME,ORBIT1(6),ORBIT2(6),PI,CJ2,CMU,PI2,PI32,SA(3),RADIAN,   CONS
000006         3ICASE,IPPIS(10),NLINE,NPAGE,INP,LINMAX,ISWTR,ISWR,NVEH,NODAY,   CONS
000007         4IV1TIM(6),IV2TIM(6),KO,TORBIT,L1DEBG(4)                CONS
000008          REAL NAME,NAME1                                         CONS
000009          LOGICAL L1DEBG                                          CONS
000010          IF(NLINES .EQ. 0) RETURN                                EJECT
000011          NLINE = NLINES                                          EJECT
000012          NPAGE = NPAGE + 1                                       EJECT
000013          WRITE(KO,9010)                                          EJECT
000014          WRITE(KO,9020)(HEADER(I),I=1,18),ICASE,NPAGE                     *NEW
000015          RETURN                                                  EJECT    **-1
000016     9010 FORMAT(1H1)                                             EJECT
000017     9020 FORMAT(1X,18A4,2X,16HLPP SIMULATION  ,5HCASE ,I4,2X,4HPAGE,I4)  *NEW
000018          END                                                     EJECT    **-1
```

```
* ECT ERRDET,1,760331, 59928    , 1
000001          SUBROUTINE ERRDET                                       ERRDET
000002    C     THIS ROUTINE CHECKS ALL ERRORS ON INPUT DATA CARDS      ERRDET
000003          COMMON /CONS/                                           CONS
000004         1ECCE,CIS,CNUV,RADIUS,HEADER(20),NAME(2),NAME1(2),TWOPI,ALPHAM,  CONS
000005         2TMES,WE,ORBTT1(6),ORBTT2(6),PI,CJ2,CMU,PI2,PI32,SA(3),RADIAN,  CONS
000006         3ICASE,IPPIS(10),NLINE,NPAGE,INP,LINMAX,ISWTR,ISWH,NVEH,NODAY,  CONS
000007         4IV1TIM(6),IV2TIM(6),KO,TORBIT,LIDEUS(4)                CONS
000008          REAL NAME,NAME1                                         CONS
000009          LOGICAL LIDEBG                                          CONS
000010          COMMON /FLAG/                                           FLAG
000011         1IFIRST,IVEH,ISTART,IEND,IDEND                           FLAG
000012          COMMON /SWAT/                                           SWAT
000013         1DLONMN(100),DLONMX(100),TIME1(100,3),DLONG1(100,3),INLAT(2),  SWAT
000014         2ISLAT(2),ISOSTR,NLAT,LATNO(100),IALT(100),ICRDI(17)     SWAT
000015          DATA I549/549/,I65/65/,I101/101/,P10/10./,I1963/63/,     SETI
000016         1R6650/6650./,R7700/7700./,RSM/.00001/,RSMI/.15/,RP65/65./,  ERRDET
000017         2R360/360./        ,I1//1/,I17/17/,I31/31/,I12/12/       ERRDET
000018          DIMENSION XTO(2),LTO(2)                                 SET2
000019          IERC = 0                                                ERRDET
000020          IF(NODAY .GE. 1 .AND. NODAY .LE.I549)GO TO 10           ERRDET
000021          CALL PAGER(2)                                           ERRDET
000022          WRITE(KO,900)                                           ERRDET
000023          IERC = IERC + 1                                         ERRDET
000024       10 IF(INLAT(1) .GE. 0  .AND. INLAT(1) .LE. INLAT(2))GO TO 15  ERRDET
000025          GO TO 30                                                ERRDET
000026       15 IF(INLAT(2) .LE. I65)GO TO 20                           ERRDET
000027          GO TO 30                                                ERRDET
000028       20 IF(ISLAT(1) .GE. 0 .AND. ISLAT(1) .LE. ISLAT(2))GO TO 25  SETI
000029          GO TO 30                                                ERRDET
000030       25 IF(ISLAT(2) .LE. I65)GO TO 40                           SETI
000031       30 CALL PAGER(3)                                           ERRDET
000032          WRITE(KO,905)                                           ERRDET
000033          IERC = IERC + 1                                         ERRDET
000034       40 ILC =INLAT(2) - INLAT(1) + 1                            ERRDET
000035          NLAT= ILC+(IABS(ISLAT(2))- IABS(ISLAT(1))+1)            ERRDET
000036          IF(NLAT.LT. I101)GO TO 45                               ERRDET
000037          CALL PAGER(2)                                           ERRDET
000038          WRITE(KO,910)                                           ERRDET
000039          IERC = IERC + 1                                         ERRDET
000040       45 DO 55 I=1,10,2                                          ERRDET
000041          J = I+1                                                 ERRDET
000042          IF(IPPIS(I) .EQ. 0 .AND. IPPIS(J) .EQ. 0) GO TO 60      ERRDET
000043          IF(IPPIS(I) .GE. 1 .AND. IPPIS(I) .LE. IPPIS(J))GO TO 50  ERRDET
000044          GO TO 55                                                ERRDET
000045       50 IF(IPPIS(J) .LE.I549)GO TO 55                           ERRDET
000046       53 CALL PAGER(3)                                           ERRDET
000047          WRITE(KO,915)I,J                                        ERRDET
000048          IERC = IERC + 1                                         ERRDET
000049       55 CONTINUE                                                ERRDET
000050       60 IF(IVEH .GE. 1.AND. IVEH .LE. 2)GO TO 65                ERRDET
000051          CALL PAGER(2)                                           ERRDET
000052          WRITE(KO,920)                                           ERRDET
000053          IERC = IERC + 1                                         ERRDET
000054       65 DO 80 I=1,3                                             ERRDET
000055          J = I+1                                                 ERRDET
000056          IF(ABS(SA(I)).GE. 0..AND.ABS(SA(I)).LE. P10)GO TO 70    ERRDET
000057          GO TO 75                                                ERRDET
000058       70 IF( I .EQ. 3)GO TO 90                                   ERRDET
```

```
000059          IF(    SA(I) .LT. SA(J))GO TO 90                                ERRDET
000060       75 CALL PAGER(2)                                                   ERRDET
000061          WRITE(KO,925)I,J                                                ERRDET
000062          IERC = IERC + 1                                                 ERRDET
000063       80 CONTINUE                                                        ERRDET
000064       90 IF(IV1TIM(1) .LE. I1963)GO TO 93                                        *NEW
000065          IF(IVEH .LT. 2)GO TO 95                                                 *NEW
000066          IF(IV2TIM(1) .GT. I1963)GO TO 95                                        *NEW
000067       93 CALL PAGER(2)                                                           *NEW
000068          WRITE(KO,930)                                                   ERRDET  **-2
000069          IERC = IERC + 1                                                 ERRDET
000070       95 IF(ORBIT1(1) .GE. R6650 .AND. ORBIT1(1) .LE. R7700)GO TO 100    ERRDET
000071          GO TO 105                                                       ERRDET
000072      100 IF(IVEH .LT. 2)GO TO 110                                                 *NEW
000073          IF(ORBIT2(1) .GE. R6650 .AND. ORBIT2(1) .LE. R7700)GO TO 110            *NEW
000074      105 CALL PAGER(2)                                                   ERRDET  **-1
000075          WRITE(KO,935)                                                   ERRDET
000076          IERC = IERC + 1                                                 ERRDET
000077      110 IF(ORBIT1(2) .GE. RSM .AND. ORBIT1(2) .LE. RSM1)GO TO 115       ERRDET
000078          GO TO 120                                                       ERRDET
000079      115 IF(IVEH .LT. 2)GO TO 125                                                 *NEW
000080          IF(ORBIT2(2) .GE. RSM .AND. ORBIT2(2) .LE. RSM1)GO TO 125               *NEW
000081      120 CALL PAGER(2)                                                   ERRDET  **-1
000082          WRITE(KO,940)                                                   ERRDET
000083          IERC = IERC + 1                                                 ERRDET
000084      125 IF(ORBIT1(4) .GE. 0 .AND. ORBIT1(4) .LE. R360)GO TO 130         ERRDET
000085          GO TO 135                                                       ERRDET
000086      130 IF(IVEH .LT. 2)GO TO 140                                                 *NEW
000087          IF(ORBIT2(4) .GE. 0..AND. ORBIT2(4) .LE. R360)GO TO 140                 *NEW
000088      135 CALL PAGER(2)                                                   ERRDET  **-1
000089          WRITE(KO,945)                                                   ERRDET
000090          IERC = IERC + 1                                                 ERRDET
000091      140 IF(ORBIT1(5) .GE. 0 .AND. ORBIT1(5) .LE. R360)GO TO 145         ERRDET
000092          GO TO 150                                                       ERRDET
000093      145 IF(IVEH .LT. 2)GO TO 155                                                 *NEW
000094          IF(ORBIT2(5) .GE. 0. .AND. ORBIT2(5) .LE. R360)GO TO 155                *NEW
000095      150 CALL PAGER(2)                                                   ERRDET  **-1
000096          WRITE(KO,950)                                                   ERRDET
000097          IERC = IERC + 1                                                 ERRDET
000098      155 IF(ABS(ORBIT1(6)).GT. RP65)GO TO 157                                     *NEW
000099          IF(IVEH .LT. 2)GO TO 160                                                 *NEW
000100          IF(ABS(ORBIT2(6)) .LE. RP65)GO TO 160                                    *NEW
000101      157 CALL PAGER(2)                                                           *NEW
000102          WRITE(KO,955)                                                   ERRDET  **-3
000103          IERC = IERC + 1                                                 ERRDET
000104      160 CONTINUE                                                                 *NEW
000105      175 IF(IV1TIM(3) .GE. 1 .AND. IV1TIM(3) .LE. I31)GO TO 180          SETT    *-10
000106          GO TO 185                                                       ERRDET
000107      180 IF(IVEH .LT. 2)GO TO 187                                                 *NEW
000108          IF(IV2TIM(3) .GE. 1 .AND. IV2TIM(3) .LE. I31)GO TO 187                  *NEW
000109      185 CALL PAGER(2)                                                   ERRDET  **-1
000110          WRITE(KO,965)                                                   ERRDET
000111          IERC = IERC + 1                                                 ERRDET
000112      187 IF(IV1TIM(2) .GE. 1 .AND. IV1TIM(2) .LE. I12)GO TO 190          ERRDET
000113          GO TO 195                                                       ERRDET
000114      190 IF(IVEH .LT. 2)GO TO 200                                                 *NEW
000115          IF(IV2TIM(2) .GE. 1 .AND. IV2TIM(2) .LE. I12)GO TO 200                  *NEW
000116      195 CALL PAGER(2)                                                   ERRDET  **-1
000117          WRITE(KO,970)                                                   ERRDET
000118          IERC = IERC + 1                                                 ERRDET
```

```
000119      200 IF(IERC .EQ. 0)GO TO 210                                      ERRDET
000120          CALL PAGEP(2)                                                 ERRDET
000121          WRITE(KU,980)                                                 ERRDET
000122          STOP                                                          ERRDET
000123      210 RETURN                                                        ERRDET
000124      900 FORMAT(1H0,54H*** NODAY IS OUT OF RANGE. IT IS NOT BETWEEN 1 AND 5ERRDET
000125         149 )                                                          ERRDET
000126      905 FORMAT(1H0,107H*** INLAT OR ISLAT IS OUT OF RANGE. THEY ARE NOT BEERRDET
000127         1TWEEN 0 AND + 65 OR THE FIRST VALUE OF EITHER ONE IS NOT/29HLESS TSFT1
000128         2HAN OR = TO THE SECOND )                                      SFT1
000129      910 FORMAT(1H0,20H*** THE TOTAL NUMBER OF LATITUDE POINTS SPECIFIED BYERRDET
000130         1 ISLAT AND INLAT EXCEEDS 100)                                 ERRDET
000131      915 FORMAT(1H0, 10H*** IPPIS(,I2,12H) AND IPPIS(,I2, 7HH) ARE OUT OF RERRDET
000132         1ANGE. THEY ARE NOT BETWEEN 1 AND 549 OR THE VALUE OF THE FIRST/   ERRDET
000133         251HENTRY IN A PAIR IS NOT LESS THAN OR = TO THE SECOND)       ERRDET
000134      920 FORMAT(1H0,31H*** IVER IS NOT BETWEEN 1 AND 2)                ERRDET
000135      925 FORMAT(1H0,62H*** EITHER AN SA ENTRY IS NOT BETWEEN 0 AND ABS(10) ERRDET
000136         1UN ENTRY (,I2,5H) IS NOT LESS THAN ENTRY(,I2,1H))             SFT1
000137      930 FORMAT(1H0,58H*** EITHER IV1TIM(1) OR IV2TIM(1) IS NOT GREATER THAERRDET
000138         1N 1963)                                                       ERRDET
000139      935 FORMAT(1H0,73H*** EITHER ORBIT1(1) OR ORBIT2(1) IS NOT BETWEEN 665ERRDET
000140         10 AND 7700 KILOMETERS)                                        ERRDET
000141      940 FORMAT(1H0,63H*** EITHER ORBIT1(2) OR ORBIT2(2) IS NOT BETWEEN .00ERRDET
000142         1001 AND .15)                                                  ERRDET
000143      945 FORMAT(1H0,58H*** EITHER ORBIT1(4) OR ORBIT2(4) IS NOT BETWEEN 0 AERRDET
000144         1ND 360)                                                       ERRDET
000145      950 FORMAT(1H0,58H*** EITHER ORBIT1(5) OR ORBIT2(5) IS NOT BETWEEN 0 AERRDET
000146         1ND 360)                                                       ERRDET
000147      955 FORMAT(1H0,59H*** EITHER ORBIT1(6) OR ORBIT2(6) IS NOT BETWEEN -65ERRDET
000148         1 AND 65)                                                      ERRDET
000149      965 FORMAT(1H0,50H*** IV1TIM(3) OR IV2TIM(3) IS NOT BETWEEN 1 AND 31) ERRDET   **-2
000150      970 FORMAT(1H0,50H*** IV1TIM(2) OR IV2TIM(2) IS NOT BETWEEN 1 AND 12) ERRDET
000151      980 FORMAT(1H0,78H*** THIS JOB IS ABANDONED DUE TO THE OCCURANCE OF 1 ERRDET
000152         1OR MORE FATAL INPUT ERRORS)                                   ERRDET
000153          END                                                           ERRDET
```

```
• FLT HECTOR,1,760331, 59920   , 1

000001              SUBROUTINE HECTOR                                          HECTOR
000002        C                                                                HECTOR
000003        C                                                                HECTOR
000004        C  SUBROUTINE HECTOR GENERATES THE EPHEMERIS DATA FOR TWO VEHICLES FOR  HECTOR
000005        C  ONE DAY.  THE EPHEMERIS IS MAINTAINED AS A SET OF ORBITAL ELEMENTS.  HECTOR
000006        C                                                                HECTOR
000007        C        INPUT --- SEMI-MAJOR AXIS (E.R.)                        HECTOR
000008        C              --- ECCENTRICITY                                  HECTOR
000009        C              --- INCLINATION (RAD)                             HECTOR
000010        C              --- LONGITUDE OF THE NODE AT REF. TIME (RAD)      HECTOR
000011        C              --- ARGUMENT OF PERIGEE PASSAGE (RAD)             HECTOR
000012        C              --- TIME OF PERIGEE PASSAGE (SEC)                 HECTOR
000013        C                          OR                                    HECTOR
000014        C              --- LATITUDE + DIRECTION OF TRAVEL                 HECTOR
000015        C                                                                HECTOR
000016        C              --- TIME OF INJECTION (YR/MO/DY/HR/MN/SEC)         HECTOR
000017        C                                                                HECTOR
000018        C        OUTPUT--- SEMI-MAJOR AXIS                               HECTOR
000019        C              --- ECCENTRICITY                                  HECTOR
000020        C              --- INCLINATION                                   HECTOR
000021        C              --- LONGITUDE OF THE NODE                         HECTOR
000022        C              --- ARGUMENT OF PERIGEE                            HECTOR
000023        C              --- TIME OF PERIGEE PASSAGE                       HECTOR
000024        C              --- START TIME OF REV                             HECTOR
000025        C              --- RIGHT ASC. OF NODE                            HECTOR
000026        C              --- PRECESSION RATE OF LINE OF APSIDES            HECTOR
000027        C              --- REGRESSION RATE OF NODE                       HECTOR
000028        C              --- ORBIT PLANE SUN ANGLE                         HECTOR
000029        C              --- ORBIT PLANE SUN ANGLE RATE                    HECTOR
000030              LOGICAL   ICHAR(4)                                         HECTOR
000031              DIMENSION IREVCT(17)                                       HECTOR
000032              DATA IREVCT/1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17/     HECTOR
000033              INTEGER   IFLAG                                            HECTOR
000034              DATA IFLAG/1/                                              HECTOR
000035        C                                                                HECTOR
000036        C                                                                HECTOR
000037              COMMON/ICARUS/ CARUS(8,17,2)                               HECTOR
000038              COMMON /ACCNUM/ ACCNUM(100)                                SETT
000039              EQUIVALENCE (ACCNUM(1),KVFH)                               SETT
000040              COMMON /CONS/                                              CONS
000041             1ECCF,CIS,GMUV,RADIUS,HEADER(20),NAME(2),NAME1(2),TWOPI,ALPHAM,  CONS
000042             2THLS,KI,ORBIT1(6),ORBIT2(6),PI,CJ2,CMU,PI2,PI32,SA(3),RADIAN,  CONS
000043             3ICASE,IPPTS(10),NLINE,NPAGE,INP,LINMAX,ISHTH,ISHR,NVEH,NODAY,  CONS
000044             4IV1TIM(6),IV2TIM(6),KO,IORBIT,LIDEBG(4)                   CONS
000045              REAL NAME,NAME1                                            CONS
000046              LOGICAL LIDEBG                                             CONS
000047        C                                                                HECTOR
000048              COMMON /HECCOM/                                            HECCOM
000049             1ALPHAT,XBETAO(2),XBETA(2,2),XNO(2),XOMEGO(2),TASCNP(2),XTO(2),  HECCOM
000050             2PERTOD(2),LMO,LDA,NREV1,NREV2,IYR,IYRB,IDAY,IPDAY,ICDAY,NREVIP,  HECCOM
000051             3NREV2P,IV1DAY,IV2DAY,LYR                                  SETT
000052              COMMON /ACCEPH/                                            ACCEPH
000053             1IPNODE(1,2),SN(1,2),EC(1,2),XIN(1,2),RTASC(1,2),WP(1,2),TNDE(1,2)  ACCEPH
000054             2,PG(1)                                                    ACCEPH
000055              EQUIVALENCE (10,ICHAR(1))                                  HECTOR
000056              DATA 10/0/                                                 HECTOR
000057        C                                                                HECTOR
000058              DIMENSION XT(2)                                            HECTOR
```

```
000059  C                                                                HECTOR
000060  C  COMPUTE YEAR-1900,MONTH,DAY(FLT. PT.) FOR INPUT TO LFPA       HECTOR
000061  C                                                                HECTOR
000062  C  TEST FOR END OF YEAR                                          HECTOR
000063  C                                                                HECTOR
000064      IF(IDAY.LT.IPDAY) LYR=LYR+1                                  HECTOR
000065  C                                                                HECTOR
000066  C  CALL CLDAY TO COMPUTE LMONTH AND LDAY                         HECTOR
000067  C                                                                HECTOR
000068      CALL CLDAY                                                   HECTOR
000069  C                                                                HECTOR
000070  C  UPDATE IPDAY FOR NEXT CALL TO HECTOR                          HECTOR
000071  C                                                                HECTOR
000072      IF(IDAY .LT. IPDAY)IPDAY = IDAY - 1                          HECTOR
000073  C                                                                SET1
000074      FLDA=LDA                                                     HECTOR
000075      CALL LFPA(FLDA,LMO,LYR,ALFAGO,DAYS)                          HECTOR
000076      IVEH=KVEH                                                    HECTOR
000077      IREV = 1                                                     HECTOR
000078      IF(ICDAY .GT. 1) GO TO 202                                   HECTOR
000079      IF(KVEH .EQ. 1)GO TO 10                                      HECTOR
000080      IF(IV2DAY .EQ. IDAY)GO TO 100                                HECTOR
000081      GO TO 202                                                    HECTOR
000082   10 IF(IV1DAY.EQ.IDAY) GO TO 100                                 HECTOR
000083      GO TO 202                                                    HECTOR
000084  100 CONTINUE                                                     HECTOR
000085      IF(KVEH .EQ. 1)GO TO 105                                     HECTOR
000086      XTO(IVEH)=IV2TIM(4)*3600+IV2TIM(5)*60+IV2TIM(6)              HECTOR
000087      GO TO 40                                                     HECTOR
000088  105 XTO(IVEH)=IV1TIM(4)*3600+IV1TIM(5)*60+IV1TIM(6)              HECTOR
000089  110 CONTINUE                                                     HECTOR
000090   40 CONTINUE                                                     HECTOR
000091  C                                                                HECTOR
000092  C  KEPLERIAN ORBIT ELEMENTS                                      HECTOR
000093  C                                                                HECTOR
000094  C          CARUS(1,1,IVEH)=SEMI-MAJOR AXIS (E.R.)                HECTOR
000095  C          CARUS(2,1,IVEH)=ECCENTRICITY (N.D.)                   HECTOR
000096  C          CARUS(3,1,IVEH)=INCLINATION (RAD)                     HECTOR
000097  C          CARUS(4,1,IVEH)=LONG. OF ASC. NODE (RAD)              HECTOR
000098  C          CARUS(5,1,IVEH)=ARG. OF PERIFOCUS (RAD)               HECTOR
000099  C          CARUS(6,1,IVEH)=TIME OF PERIFOCUS PASSAGE (SEC)       HECTOR
000100  C          CARUS(7,1,IVEH)=TIME AT START OF THE REV (SEC)        HECTOR
000101  C          CARUS(8,1,IVEH)=RIGHT ASCENSION OF THE NODE (RAD)     HECTOR
000102  C          CARUS(9,1,IVEH) (NOT USED)                            HECTOR
000103  C          CARUS(10,1,IVEH) (NOT USED)                           HECTOR
000104  C                                                                HECTOR
000105  C  COMPUTE ORBITAL ELEMENTS FOR FIRST REV                       HECTOR
000106  C                                                                HECTOR
000107  C  MEAN MOTION                                                   HECTOR
000108  C                                                                HECTOR
000109      XN = SQRT(CMU/CARUS(1,1,IVEH)**3)                            HECTOR
000110      P = TWOPI/XN                                                 HECTOR
000111      KP = 1                                                       HECTOR
000112   50 XTEM      = XTO(IVEH) -KP*P                                  HECTOR
000113      IF(XTEM      .LT. 0.)GO TO 60                                HECTOR
000114      KP = KP + 1                                                  HECTOR
000115      GO TO 50                                                     HECTOR
000116   60 KP = KP - 1                                                  HECTOR
000117      XTO(IVEH) = XTEM      +P                                     HECTOR
000118      DELOMG = PI*P/43200.                                         HECTOR
                                                                         *NEW
                                                                         *NEW
                                                                         *NEW
                                                                         *NEW
                                                                         *NEW
                                                                         *NEW
                                                                         *NEW
                                                                         *NEW
                                                                         *NEW
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
      CARUS(4,1,IVEH) = CARUS(4,1,IVEH) + KP*DELOMG                     *NEW
      IF(CARUS(4,1,IVEH) .LT. TWOPI)GO TO 70                            *NEW
      CARUS(4,1,IVEH) = CARUS(4,1,IVEH) - DELOMG                        *NEW
      KP = KP + 1                                                       *NEW
   70 CONTINUE                                                          *NEW
      XECC = CARUS(2,1,IVEH)
C  FORM (1.-ECC**2) + (1.-ECC**2)**1/2                                  HECTOR
      TERM1 = 1. - XECC**2                                              HECTOR
      TERM2 = SQRT(TERM1)                                               HECTOR
C                                                                       HECTOR
C  TIME OF PERIGEE PASSAGE INPUT: IF NOT,COMPUTE INITIAL VALUE FROM LATIHECTOR
C                                                                       HECTOR
      IF(IORBIT.EQ.0) GO TO 150                                         HECTOR
C                                                                       HECTOR
C  COMPUTE TIME OF PERIGEE PASSAGE FROM LATITUDE                        HECTOR
C                                                                       HECTOR
      XLATI=CARUS(6,1,IVEH)                                             HECTOR
      XI=CARUS(3,1,IVEH)                                                HECTOR
      XWI = ASIN(SIN(XLATI)/SIN(XI))                                    HECTOR
C                                                                       HECTOR
C                                                                       HECTOR
C  INPUT TYPE FLAG                                                      HECTOR
C       FLAG            INPUT TYPE                                      HECTOR
C        0          TIME OF PERIGEE PASSAGE (FOR BOTH VEHICLES)         HECTOR
C        1          LAT. OF INJECTION INPUT - ASC.  (BOTH VEH.)         HECTOR
C        2          LAT. OF INJECTION INPUT - DESC. (BOTH VEH.)         HECTOR
C        3          LAT. OF INJECTION INPUT - ASC.-VEH 1 + DES.-VEH 2   HECTOR
C        4          LAT. OF INJECTION INPUT - DES.-VEH 1 + ASC.-VEH 2   HECTOR
C                                                                       HECTOR
C  ASCENDING                                                            HECTOR
C                                                                       HECTOR
      IF (IORBIT .EQ. 1 .AND. XLATI .LT. 0.) XWI = TWOPI + XWI          HECTOR
      IF(IORBIT.EQ.3.AND.IVEH.EQ.1.AND.XLATI.LT.0.) XWI=TWOPI+XWI       HECTOR
      IF(IORBIT.EQ.4.AND.IVEH.EQ.2.AND.XLATI.LT.0.) XWI=TWOPI+XWI       HECTOR
C                                                                       HECTOR
C  DESCENDING                                                           HECTOR
C                                                                       HECTOR
      IF(IORBIT.EQ.2) XWI=PI-XWI                                        HECTOR
      IF(IORBIT.EQ.3.AND.IVEH.EQ.2) XWI = PI-XWI                        HECTOR
      IF(IORBIT.EQ.4.AND.IVEH.EQ.1) XWI = PI-XWI                        HECTOR
      XNUI=XWI-CARUS(5,1,IVEH)                                          HECTOR
C                                                                       HECTOR
C  TEST FOR NU LESS THAN ZERO                                           HECTOR
C                                                                       HECTOR
      IF(XNUI .LT. 0.)XNUI = TWOPI + XNUI                               HECTOR
      COSEI=(1./XECC)*(1.-TERM1/(1.+XECC*COS(XNUI)))                    *NEW
      XEI = ASIN(SIN(XNUI)*(1.-XECC*COSEI)/TERM2)                       HECTOR **-1
      IF(COSEI .LT. 0.)XEI = PI - XEI                                   HECTOR
      IF(XEI .LT. 0. .AND. COSEI .GT. 0.)XEI = TWOPI + XEI              *NEW
      DELTAU=(XEI-XECC*SIN(XEI))/XN                                     *NEW
      CARUS(6,1,IVEH)=XTO(IVEH)-DELTAU                                  HECTOR
  150 CONTINUE                                                          HECTOR
C  SET PERIGEE PASSAGE TIME                                             HECTOR
C                                                                       HECTOR
      XT = CARUS(6,1,IVEH)                                              HECTOR
C                                                                       HECTOR
C                                                                       HECTOR
C  MEAN ANOMALY                                                         HECTOR
C                                                                       HECTOR
      XN=XNI*(XTO(IVEH)-CARUS(6,1,IVEH))                                HECTOR
```

```
000179          C                                                                HECTOR
000180          C  CALL KEPLER TO COMPUTE ECC. ANOMALY                           HECTOR
000181          C                                                                HECTOR
000182              CALL KEPLER(XM,XECC,XF,ERROR)                                HECTOR
000183              IF(ERROR.EQ.0.) GO TO 120                                    HECTOR
000184              CALL PAGER(2)                                                HECTOR
000185              WRITE(KO,1000)                                               HECTOR
000186         1000 FORMAT(1H0,80H** ERROR (KEPLER) SOLUTION FOR ECC. ANOMALY DID NOT HECTOR
000187             1CONVERGE AFTER 50 ITERATIONS )                               HECTOR
000188          C                                                                HECTOR
000189          C                                                                HECTOR
000190          120 CONTINUE                                                     HECTOR
000191              SINE = SIN(XE)                                               HECTOR
000192              COSE = COS(XE)                                               HECTOR
000193          C                                                                HECTOR
000194          C  FORM (1.-ECC*COS(E))                                          HECTOR
000195          C                                                                HECTOR
000196              TERM3 = 1. - XECC*COSE                                       HECTOR
000197          C                                                                HECTOR
000198          C  TRUE ANOMALY (INJECTION)                                      HECTOR
000199          C                                                                HECTOR
000200              XNU = ASIN(TERM2*SINE/TERM3)                                 HECTOR
000201          C                                                                HECTOR
000202          C  TEST FOR QUADRANT OF NU                                       HECTOR
000203          C                                                                HECTOR
000204              COSNU = (COSE-XECC)/(1.-XECC*COSE)                           HECTOR
000205          C                                                                HECTOR
000206              IF(COSNU .LT. 0.) XNU = PI - XNU                             HECTOR
000207              IF(XNU .LT. 0. .AND. COSNU .GT. 0.) XNU = TWOPI + XNU        HECTOR
000208          C                                                                HECTOR
000209          C  OTHERWISE NU IS UNCHANGED....                                 HECTOR
000210          C                                                                HECTOR
000211          C  INPLANE ANGLE FROM ASC. NODE TO INJ. POINT                    HECTOR
000212          C                                                                HECTOR
000213              XU = PIMOD(CARUS(5,1,IVEH)+XNU)                              HECTOR
000214          C                                                                HECTOR
000215          C  INPLANE ANGLE FROM INJ. POINT TO START OF NEXT REV.(SOUTHERNMOST PT.)HECTOR
000216          C                                                                HECTOR
000217              DXNU = 3. * PI2 - XU                                         HECTOR
000218          C                                                                HECTOR
000219          C  TRUE ANOMALY AT S. P.                                         HECTOR
000220          C                                                                HECTOR
000221              XNU1 = PIMOD(XNU + DXNU)                                     HECTOR
000222          C                                                                HECTOR
000223          C  COSINE OF E AT NU                                             HECTOR
000224          C                                                                HECTOR
000225              COSE1 = (1./XECC)*(1.-TERM1/(1.+XECC*COS(XNU1)))             HECTOR
000226          C                                                                HECTOR
000227          C  ECC. ANOMALY AT NU1                                           HECTOR
000228          C                                                                HECTOR
000229              XE1 = ASIN(SIN(XNU1)*(1.-XECC*COSE1)/TERM2)                  HECTOR
000230              IF(COSE1 .LT. 0.) XE1 = PI - XE1                             HECTOR
000231              IF(XE1 .LT. 0. .AND. COSE1 .GT. 0.) XE1 = TWOPI + XE1        HECTOR
000232          C                                                                HECTOR
000233          C  NO. OF PERIGEE - PERIGEE REVS                                 HECTOR
000234          C                                                                HECTOR
000235              K = (XNU + DXNU)/TWOPI                                       HECTOR
000236              XK = K                                                       HECTOR
000237          C                                                                HECTOR
000238              TERM4 = TWOPI/XN                                             HECTOR
```

```
      PERIOD(IVEH) = TERM4                                              HECTOR
C                                                                       HECTOR
C  TIME AT S. P. (1-S)                                                  HECTOR
C                                                                       HECTOR
      CARUS(7,1,IVEH)=(XE1-XECC*SIN(XE1))/XN+XT+XK*TERM4                HECTOR
C                                                                       HECTOR
C  TRUE ANOMALY AT ASCENDING NODE                                       HECTOR
C                                                                       HECTOR
      XNUNDE = XNU1 + PI2                                               HECTOR
C                                                                       HECTOR
C  ECCENTRIC ANOMALY AT ASCENDING NODE                                  HECTOR
C                                                                       HECTOR
      SINEND = SIN(XNUNDE)*(TERM1/(1.+XECC*COS(XNUNDE)))/TERM2          HECTOR
      XESTAR = ACOS((1.-TERM1/(1.+XECC*COS(XNUNDE)))/XECC)              HECTOR
      IF (SINEND .LT. 0.0) XESTAR = TWOPI - XESTAR                      HECTOR
C                                                                       HECTOR
C  TIME OF ASC. NODE FOR REV 1                                          HECTOR
C                                                                       HECTOR
      XT1(IVEH) = (XESTAR-XECC*SIN(XESTAR))/XN+XT+XK*TERM4              HECTOR
  140 IF( XT1(IVEH) .GE. CARUS(7,1,IVEH)) GO TO 145                     HECTOR
      XT1(IVEH) = XT1(IVEH) + TERM4                                     HECTOR
      GO TO 140                                                         HECTOR
C                                                                       HECTOR
  145 CONTINUE                                                          HECTOR
C                                                                       HECTOR
      TNNODE(1,IVEH) = XT1(IVEH)                                        HECTOR
C                                                                       HECTOR
C  RECESSION OF NODE + PREC. OF LINE OF APSIDES                         HECTOR
C                                                                       HECTOR
      TERM5 = (-3./2.)*(XN*CJ2)/((CARUS(1,1,IVEH)**2)*TERM1**2)         HECTOR
      XOMEGD(IVEH) = TERM5*COS(CARUS(3,1,IVEH))                         HECTOR
      XWD(IVEH)=TERM5*(5.*SIN(CARUS(3,1,IVEH))**2/2.-2.)                HECTOR
C                                                                       HECTOR
C  LONG. OF NODE + ARG. OF PERIGEE AT TIME OF NODE OF FIRST REV         HECTOR
C                                                                       HECTOR
      CARUS(4,1,IVEH)=PIMOD(CARUS(4,1,IVEH)+(XOMEGD(IVEH)-WE)*(XT1(IVEH)HECTOR
     1                -XT0(IVEH)))                                      HECTOR
      CARUS(5,1,IVEH)=PIMOD(CARUS(5,1,IVEH)+XWD(IVEH)*(XT1(IVEH)-XT0(IVEHECTOR
     *H)))                                                              HECTOR
      IF(CARUS(5,1,IVEH).GT.PI32) CARUS(5,1,IVEH) = CARUS(5,1,IVEH)     HECTOR
     *                                              -TWOPI              HECTOR
C                                                                       HECTOR
C  TIME OF PERIGEE PASSAGE FOR FIRST REV                                HECTOR
C                                                                       HECTOR
      CARUS(6,1,IVEH)=CARUS(6,1,IVEH)+(XK+1.)*TERM4                     HECTOR
C                                                                       HECTOR
C  RIGHT ASCENSION OF NODE AT TIME OF NODAL CROSSING                    HECTOR
C                                                                       HECTOR
      CARUS(8,1,IVEH)=PIMOD(ALFA0+WE*XT1(IVEH)+CARUS(4,1,IVEH))         HECTOR
C                                                                       HECTOR
C  CALL SUBROUTINE ALPHA TO COMPUTE RIGHT ASC. OF SUN AT TIME OF 1ST REVHECTOR
C                                                                       HECTOR
      CALL ALPHA(IFLAG)                                                 HECTOR
C                                                                       HECTOR
C     -ORBIT PLANE SUN ANGLE -- BETA                                    HECTOR
C                                                                       HECTOR
      XBETA(1,IVEH)=PIMOD(ALPHAT-CARUS(8,1,IVEH))                       HECTOR
C                                                                       HECTOR
C     -ORBIT PLANE SUN ANGLE -- BETA            0 HOUR                  HECTOR
C                                                                       HECTOR
```

```
      XOMGPR = CARUS(4,1,IVEH)+(XOMEGD(IVEH)-WE)*(-XT1(IVEH))          HECTOR
      XBETA(2,IVEH) = PI-XOMGPR                                        HECTOR
      IF(XOMGPR.GT.PI) XBETA(2,IVEH) = 3.*PI-XOMGPR                    HECTOR
C                                                                      HECTOR
C     -RATE OF CHANGE OF ANGLE -- BETA-DOT                             HECTOR
C                                                                      HECTOR
      XBETAD(IVEH) = (XBETA(1,IVEH)-XBETA(2,IVEH))/XT1(IVEH)           HECTOR
C                                                                      HECTOR
C                                                                      HECTOR
      GO TO 211                                                        HECTOR
C                                                                      HECTOR
C                                                                      HECTOR
C  SET ELEMENT                                                         HECTOR
C                                                                      HECTOR
  202 CONTINUE                                                         HECTOR
C                                                                      HECTOR
C  SET REV COUNT FROM PREVIOUS DAY                                     HECTOR
C                                                                      HECTOR
      NREVP = NREV1P                                                   HECTOR
      IF(IVEH.EQ.2) NREVP=NREV2P                                       HECTOR
      CARUS(1,1,IVEH) = CARUS(1,NREVP,IVEH)                            HECTOR
      XN = SQRT(GMU/CARUS(1,1,IVEH)**3)                                HECTOR
      CARUS(2,1,IVEH) = CARUS(2,NREVP,IVEH)                            HECTOR
      CARUS(3,1,IVEH) = CARUS(3,NREVP,IVEH)                            HECTOR
      TERM4 = PERIOD(IVEH)                                             HECTOR
      CARUS(4,1,IVEH) = PIMOD(CARUS(4,NREVP,IVEH)+(XOMEGD(IVEH)-WE)    HECTOR
     1                 *TERM4)                                         HECTOR
      CARUS(5,1,IVEH) = PIMOD(CARUS(5,NREVP,IVEH)+XWD(IVEH)            HECTOR
     1                 *TERM4)                                         HECTOR
      IF(CARUS(5,1,IVEH).GT.PI32) CARUS(5,1,IVEH) = CARUS(5,1,IVEH)    HECTOR
     *                                              -TWOPI             HECTOR
C                                                                      HECTOR
C  TEST FOR SECOND DAY OF LAUNCH FOR EITHER VEHICLE                    HECTOR
C                                                                      HECTOR
      IF(ICDAY .GT. 2) GO TO 206
      IF(IVEH.EQ.2) GO TO 204                                          HECTOR
      IF((IDAY-IV1DAY)-1) 206,205,206                                  HECTOR
  204 IF((IDAY-IV2DAY)-1) 206,205,206                                  HECTOR
C  COMPUTE TIME OF PERIGEE PASSAGE FOR DAY 2                           HECTOR
  205 CARUS(6,1,IVEH) = CARUS(6,NREVP,IVEH)+(TWOPI+(XWD(IVEH)*(86400.  HECTOR
     1                 -XT0(IVEH))))/XN-86400.                         HECTOR
      GO TO 208                                                        HECTOR
  206 CONTINUE                                                         HECTOR
C  COMPUTE TIME OF PERIGEE PASSAGE FOR DAY GREATER THAN DAY 2 OF MISSIONHECTOR
      CARUS(6,1,IVEH) = CARUS(6,NREVP,IVEH)+(TWOPI+ XWD(IVEH)*86400.)/XNHECTOR
     1                 -86400.                                         HECTOR
  208 CONTINUE                                                         HECTOR
      CARUS(7,1,IVEH) = CARUS(7,NREVP,IVEH)+TERM4-86400.               HECTOR
C                                                                      HECTOR
C  COMPUTE TIME OF ASC. NODE FOR FIRST REV OF DAY OTHER THAN DAY 1     HECTOR
C                                                                      HECTOR
      XT1(IVEH) = (TASCNP(IVEH)+NREVP*TERM4)-86400.                    HECTOR
C                                                                      HECTOR
      TMNODE(1,IVEH) = XT1(IVEH)                                       HECTOR
C                                                                      HECTOR
C  COMPUTE RT. ASC. OF NODE AT T1 FOR DAY OTHER THAN DAY 1             HECTOR
      CARUS(8,1,IVEH) = PIMOD(ALPHA0+WE*XT1(IVEH)+CARUS(4,1,IVEH))     HECTOR
  210 CONTINUE                                                         HECTOR
C                                                                      HECTOR
C  CALCULATE BETA-DOT FOR DAY OTHER THAN LAUNCH                        HECTOR
```

```
000359          C                                                       HECTOR
000360          C  BETA FOR DAY = 1                                     HECTOR
000361          C                                                       HECTOR
000362              XBETA(2,IVEH)=XBETA(1,IVEH)                         HECTOR
000363              CALL ALPHA(IFLAG)                                   HECTOR
000364          C                                                       HECTOR
000365          C  BETA FOR THIS DAY                                    HECTOR
000366          C                                                       HECTOR
000367              XBETA(1,IVEH)=PIMOD(ALPHAT-CARUS(8,1,IVEH))         HECTOR
000368          C                                                       HECTOR
000369          C  BETA DOT                                             HECTOR
000370          C                                                       HECTOR
000371              XBETAD(IVEH)=(XBETA(1,IVEH)-XBETA(2,IVEH))/(XTI(IVEH)  HECTOR
000372             1              +(86400.-TASCNP(IVEH)))              HECTOR
000373          C                                                       HECTOR
000374          C  COMPUTE ORBIT ELEMENT FOR NREV .GT. 1                HECTOR
000375          C                                                       HECTOR
000376      211 DO 220 IREV=2,17                                        HECTOR
000377              IREVM1 = IREV-1                                     HECTOR
000378              TMNODE(IREV,IVEH) = TMNODE(IREVM1,IVEH)+TERM4       HECTOR
000379              CARUS(1,IREV,IVEH) = CARUS(1,IREVM1,IVEH)           HECTOR
000380              CARUS(2,IREV,IVEH) = CARUS(2,IREVM1,IVEH)           HECTOR
000381              CARUS(3,IREV,IVEH) = CARUS(3,IREVM1,IVEH)           HECTOR
000382              CARUS(4,IREV,IVEH) = PIMOD(CARUS(4,IREVM1,IVEH)+(XOMEGD(IVEH)-WE)  HECTOR
000383             1                     *TERM4)                        HECTOR
000384              CARUS(5,IREV,IVEH) = PIMOD(CARUS(5,IREVM1,IVEH)+XWD(IVEH)  HECTOR
000385             1                     *TERM4)                        HECTOR
000386              IF(CARUS(5,IREV,IVEH).GT.PI52) CARUS(5,IREV,IVEH)   HECTOR
000387             * = CARUS(5,IREV,IVEH)-TWOPI                         HECTOR
000388              CARUS(6,IREV,IVEH) = CARUS(6,IREVM1,IVEH)+TERM4     HECTOR
000389              CARUS(7,IREV,IVEH) = CARUS(7,IREVM1,IVEH)+TERM4     HECTOR
000390              CARUS(8,IREV,IVEH) = PIMOD(CARUS(8,IREVM1,IVEH)+XOMEGD(IVEH)  HECTOR
000391             1                     *TERM4)                        HECTOR
000392              IF (TMNODE(IREV,IVEH).GT.86400.) GO TO 300          HECTOR
000393      220 CONTINUE                                                HECTOR
000394              GO TO 310                                           HECTOR
000395      300 IREV = IREVM1                                           HECTOR
000396      310 CONTINUE                                                HECTOR
000397          C                                                       HECTOR
000398          C  SET REV COUNT                                        HECTOR
000399          C                                                       HECTOR
000400              IF(IVEH.EQ.1) NREV1=IREV                            HECTOR
000401              IF(IVEH.EQ.2) NREV2=IREV                            HECTOR
000402          C                                                       HECTOR
000403          C  SET TIME OF ASC. NODE (FIRST) OF THIS DAY FOR NEXT CALL TO HECTOR  HECTOR
000404          C                                                       HECTOR
000405          C                                                       HECTOR
000406              TASCNP(IVEH) = XTI(IVEH)                            HECTOR
000407      400 CONTINUE                                                HECTOR
000408          C                                                       HECTOR
000409              RETURN                                              HECTOR
000410              END                                                 HECTOR
```

```
* FL: INPROC,1,760331, 59922   , 1

000001          SUBROUTINE INPROC                                          INPROC
000002     C    THIS ROUTINE READS THE CARD INPUT,STORES THE DATA,AND PERFORMS INI INPROC
000003     C    TIAL CALCULATIONS.                                         INPROC
000004          COMMON /CONS/                                              CONS
000005         1ECCE,CIS,CNUV,RADIUS,HEADER(20),NAME(2),NAME1(2),TWOPI,ALPHAM, CONS
000006         2THES,PE,ORBIT1(6),ORBIT2(6),PI,CJ2,CMU,PI2,PI32,SA(3),RADIAN, CONS
000007         3ICASE,IPPIS(10),NLINE,NPAGE,INP,LINMAX,ISNTR,ISER,NVEH,NUDAY, CONS
000008         4IV1TIM(6),IV2TIM(6),KO,IORBIT,LIDEBG(8)                   CONS
000009          REAL NAME,NAME1                                            CONS
000010          LOGICAL LIDEBG                                             CONS
000011          COMMON /FLAG/                                              FLAG
000012         1IFIRST,IVEH,ISTART,IEND,IDEND                              FLAG
000013          COMMON /SWAT/                                              SWAT
000014         1DLONMN(100),DLONMX(100),ITMF1(100,3),DLONG1(100,3),INLAT(2), SWAT
000015         2ISLAT(2),ISOSTR,NLAT,LATHO(100),IALT(100),IGRD1(17)        SWAT
000016          DIMENSION INLATT(2),ISLATT(2),IPPIST(10)                   INPROC
000017          READ(INP,900)(HEADER(I),I=1,20)                            INPROC
000018          READ(INP,905)ICASE,IVENT,NUDAYT,(INLATT(I),I=1,2),(ISLATT(I),I=1,2INPROC
000019         1),(LIDEBG(I),I=1,8),(IPPIST(I),I=1,10),ISOSTR,IDUM         INPROC
000020          READ(INP,910)(SA(I),I=1,3),(IV1TIM(I),I=1,6),(IV2TIM(I),I=1,6), INPROC  **-1
000021         1IDUM                                                       INPROC
000022          READ(INP,915)(ORBIT1(I),I=1,6),(ORBIT2(I),I=1,6),IDUM      INPROC
000023          IF(NUDAYT .NE. 0)NUDAY = NUDAYT                            INPROC
000024          IF(IVENT .NE. 0)IVEH = IVENT                               INPROC
000025          IF(INLATT(1) .NE. 0)INLAT(1) = INLATT(1)                   INPROC
000026          IF(INLATT(2) .NE. 0)INLAT(2) = INLATT(2)                   INPROC
000027          IF(ISLATT(1) .NE. 0)ISLAT(1) = ISLATT(1)                   INPROC
000028          IF(ISLATT(2) .NE. 0)ISLAT(2) = ISLATT(2)                   INPROC
000029          DO 5 I=1,10                                                INPROC
000030          IF(IPPIST(I) .NE. 0)IPPIS(I) = IPPIST(I)                   INPROC
000031        5 CONTINUE                                                   INPROC
000032          WRITE(KO,930)(HEADER(I),I=1,18)                                *NEW
000033          WRITE(KO,935)ICASE,IVEH,NUDAY,INLAT,ISLAT,LIDEBG,IPPIS,ISOSTR  *NEW
000034          WRITE(KO,940)SA,IV1TIM,IV2TIM                                  *NEW
000035          WRITE(KO,945)ORBIT1(1),ORBIT1(2),(ORBIT1(I),I=4,6),ORBIT2(1),  *NEW
000036         1ORBIT2(2),(ORBIT2(I),I=4,6)                                    *NEW
000037          CALL EJECT(2)                                                  *NEW
000038          CALL ERROFI                                                INPROC
000039          ORBIT1(1)  = ORBIT1(1)/RADIUS                              INPROC
000040          ORBIT2(1)  = ORBIT2(1)/RADIUS                              INPROC
000041          DO 10 I=3,6                                                INPROC
000042          ORBIT1(I) = ORBIT1(I)/RADIAN                               INPROC
000043          ORBIT2(I) = ORBIT2(I)/RADIAN                               INPROC
000044       10 CONTINUE                                                   INPROC
000045          DO 15 I=1,3                                                INPROC
000046          SA(I) = SA(I)/RADIAN                                       INPROC
000047       15 CONTINUE                                                   INPROC
000048          NVEH = IVEH                                                INPROC
000049          IORBIT = 2                                                 INPROC
000050          TEMP = (-4.*PI*(ORBIT1(1))**2)*(1. - ORBIT1(2)**2)**2      INPROC
000051          TEMP1 = 3.* SQRT(CMU/(ORBIT1(1))**3)*365.25*86400.*CJ2     INPROC
000052          TEMP2 = TEMP/TEMP1                                         SET1
000053          ORBIT1(3) = ATAN(SQRT(1.-TEMP2**2)/TEMP2) + PI             SET1
000054          ORBIT1(4) = ORBIT1(4) + PI -ASIN(-TAN(ORBIT1(6))/TAN(ORBIT1(3)))  *NEW
000055          IF(ORBIT1(4) .GT. TWOPI)ORBIT1(4) = ORBIT1(4) - TWOPI          *NEW
000056          XT01 = IV1TIM(4)*3600 + IV1TIM(5)*60 + IV1TIM(6)               *NEW
000057          LT01 = XT01 - 43200. + ORBIT1(4)*43200./PI                     *NEW
000058          IF(LT01 .GE. 25200 .AND. LT01 .LE. 61200)GO TO 100             *NEW
```

```
000059          95 CALL PAGEP(2)                                                    *NEW
000060             WRITE(KO,960)                                                    *NEW
000061             STOP                                                             *NEW
000062         100 IF(IVEH .LT. 2)GO TO 110                                         *NEW
000063             TEMP = (-4.*PI*(ORBIT2(1))**2)*(1. - ORBIT2(2)**2)**2            INPROC
000064             TEMP1 = 3.* SQRT(CMU/(ORBIT2(1))**3)*365.25*86400.*CJ2           INPROC
000065             TEMP2 = TEMP/TEMP1                                               SET1
000066             ORBIT2(3) = ATAN(SQRT(1.-TEMP2**2)/TEMP2) + PI                   SET1
000067             ORBIT2(4) = ORBIT2(4) + PI -ASIN(-TAN(ORBIT2(6))/TAN(ORBIT2(3)))  *NEW
000068             IF(ORBIT2(4) .GT. TWOPI)ORBIT2(4) = ORBIT2(4) - TWOPI            *NEW
000069             XIO2 = IV2TIM(4)*3600 + IV2TIM(5)*60 + IV2TIM(6)                 *NEW
000070             LTO2 = XIO2 - 43200. + ORBIT2(4)*43200/PI                        *NEW
000071             IF(LTO2 .GE. 25200 .AND. LTO2 .LE. 61200)GO TO 110               *NEW
000072             GO TO 95                                                         *NEW
000073         110 LATNO(1) = INLAT(2)                                              *NEW
000074             DO 600 I=2,66                                                    INPROC  **-1
000075             IK = I                                                           INPROC
000076             ITEM     = LATNO(I-1) - 1                                        INPROC
000077             IF(ITEM .LE. INLAT(1))GO TO 610                                  INPROC
000078             LATNO(I) = ITEM                                                  INPROC
000079         600 CONTINUE                                                         INPROC
000080         610 LATNO(IK) = INLAT(1)                                             INPROC
000081             IK = IK + 1                                                      INPROC
000082             LATNO(IK) = -ISLAT(1)                                            SET1
000083             DO 620 I=1,65                                                    INPROC
000084             ITEM  = LATNO(I+IK-1) - 1                                        INPROC
000085             IKI = I                                                          INPROC
000086             IF(ITEM .LE. - ISLAT(2))GO TO 630                                SET1
000087             LATNO(I+IK) = ITEM                                               INPROC
000088         620 CONTINUE                                                         INPROC
000089         630 LATNO(IK+IKI) = -ISLAT(2)                                        SET1
000090             RETURN                                                           INPROC
000091         900 FORMAT(20A4)                                                     INPROC
000092         905 FORMAT(I4,I2,I3,4I2,4I1,10I3,I2,25X,I2)                          INPROC
000093         910 FORMAT(3F5.1,12I2,32X,I2)                                        *NEW
000094         915 FORMAT(2(F6.1,F6.5,4F4.0),22X,I2)                                INPROC  **-1
000095         930 FORMAT(1H1,56X,6HHEADER/1H ,5X,18A4)                             *NEW
000096         935 FORMAT(1H0,5X,95HTCASE  IVEH  NODAY  INLAT   ISLAT   LIDEBG      *NEW
000097            1     IPRTS START/STOP TIMES                ISOSTR/               *NEW
000098            17X,I4,3X,I2,4X,I5,3X,                                             *NEW
000099            24(I2,2X),1X,4I1,3X,5(I3,1X,I3,2X),2X,I2,5H   01)                  *NEW
000100         940 FORMAT(1H0,13X,95HSA                IVITIM                IV2TIM/6X*  *NEW
000101            1,3(F5.1,2X),1X,6(I2,1X),2X,6(I2,1X),36X,2H02)                     *NEW
000102         945 FORMAT(1H0,21X, 6HORBIT1,30X,6HORBIT2/6X,  F6.1,2X,F6.5,2X,       *NEW
000103            15(F4.0,2X),1X,                                                    *NEW
000104            1 F6.1,2X,F6.5,2X,3(F4.0,2X),27X,2H03)                             *NEW
000105         960 FORMAT(1H0,96H*** IVITIM(4) OR IV2TIM(4) SPECIFIES A LOCAL VEHICLEF  *NEW
000106            1 PASSAGE TIME NOT BETWEEN 700 AND 1700 HOURS)                     *NEW
000107             END                                                              INPROC
```

ORIGINAL PAGE IS OF POOR QUALITY

```
* FLT 10,1,750925, S8805 ,11
000001          UHUF
000002          DR 8,RL=1500
000003          DR 9,RL=1000
```

```
* FT KEPLER,1,750925, 58787   , 1
000001          SUBROUTINE KEPLER(XM,XECC,XE,ERROR)                                  KEPLER
000002     C                                                                         KEPLER  **-R
000003     C                                                                         KEPLER
000004     C  THIS SUBROUTINE SOLVES KEPLER'S EQUATIONS FOR ECCENTRIC ANOMALY (E)    KEPLER
000005     C  GIVEN ECCENTRICITY (ECC) AND MEAN ANOMALY (M).                         KEPLER
000006     C  THE EQUATIONS FOR THE ELLIPTIC CASE IS ---- M=E-ECC*SIN(E).             KEPLER
000007     C                                                                         KEPLER
000008     C  LOGIC FOR THE SOLUTION OF THE HYBERBOLIC CASE HAS BEEN REMOVED.        KEPLER
000009     C                                                                         KEPLER
000010     C  INPUT                                                                  KEPLER
000011     C          XM ---- MEAN ANOMALY (RAD)                                     KEPLER
000012     C          XECC -- ORBIT ECCENTRICITY (N.D.)                              KEPLER
000013     C  OUTPUT                                                                 KEPLER
000014     C          XE ---- ECCENTRIC ANOMALY (RAD)                                KEPLER
000015     C          ERROR -1 ERROR FLAG INDICATING CONVERGENCE CONDITIONS.         KEPLER
000016     C                =0 IMPLIES CONVERGENCE HAS OCCURRED.                     KEPLER
000017     C                                                                         KEPLER
000018          DIMENSION B(4),C(4)                                                  KEPLER
000019          COMMON /CONS/                                                        CONS
000020         1ECCF,C15,CNUV,RADIUS,HEADER(20),NAME(2),NAME1(2),TWOPI,ALPHAM,       CONS
000021         2THES,XE,ORBIT1(6),ORBIT2(6),PI,CJ2,CMU,PI2,PI32,SA(3),RADIAN,        CONS
000022         3ICASE,IPPIS(10),NLINE,NPAGE,INP,LINMAX,ISWTH,ISWR,NVEH,NUDAY,        CONS
000023         4IV1TIM(6),IV2TIM(6),KO,IORBIT,LIDEBG(4)                              CONS
000024          REAL NAME,NAME1                                                      CONS
000025          LOGICAL LIDEBG                                                       KEPLER
000026          REAL    M                                                            KEPLER
000027          M = XM                                                               KEPLER
000028          ECC = XECC                                                           KEPLER
000029          ERROR = 0.                                                           KEPLER
000030          IF(M.NE.0.0) GO TO 9                                                 KEPLER
000031          E = 0.                                                               KEPLER
000032          GO TO 8                                                              KEPLER
000033        9 CONTINUE                                                             KEPLER
000034     C                                                                         KEPLER
000035     C    CALCULATE INITIAL VALUE OF ECCENTRIC ANOMALY                         KEPLER
000036     C                                                                         KEPLER
000037          TEMP = M                                                             KEPLER
000038          M = PIMOD(TEMP)                                                      KEPLER
000039     C                                                                         KEPLER
000040     C    FOR B(I) .LT. M .LE. B(I+1), I=0,1,2,3, THEN E0 = C(I+1)             KEPLER
000041     C    IF M .GT. B(4), THEN E0 = (M-2*PI*ECC)/(1-ECC)        PI=3.14159..KEPLER
000042     C        WHERE      B(0) IS MINUS INFINITY                               KEPLER
000043     C                   B(1) = 1-ECC          C(1) = M/B(1)                  KEPLER
000044     C                   B(2) = PI-1-ECC       C(2) = M+ECC                   KEPLER
000045     C                   B(3) = PI+1+ECC       C(3) = (M+ECC*PI)/(1+ECC)      KEPLER
000046     C                   B(4) = 2*PI-1+ECC     C(4) = (M-ECC)                 KEPLER
000047          B(1) = 1.-ECC                                                        KEPLER
000048          B(2) = PI-1.-ECC                                                     KEPLER
000049          B(3) = PI+1.+ECC                                                     KEPLER
000050          B(4) = 2.*PI-1.+ECC                                                  KEPLER
000051          C(1) = M/B(1)                                                        KEPLER
000052          C(2) = M+ECC                                                         KEPLER
000053          C(3) = (M+PI*ECC)/(1.+ECC)                                           KEPLER
000054          C(4) = M-ECC                                                         KEPLER
000055          DO 2 J=1,4                                                           KEPLER
000056          IF(M.GT.B(J)) GO TO 2                                                KEPLER
000057          DLDF=B(J)                                                            KEPLER
000058          GO TO 3                                                              KEPLER
```

```
000059          2 CONTINUE                                              KEPLER
000060            OLDF=(M-2.0*PI*ECC)/(1.-ECC)                          KEPLER
000061      C                                                           KEPLER
000062      C     CALCULATE ECCENTRIC ANOMALY                           KEPLER
000063      C                                                           KEPLER
000064          3 CONTINUE                                              KEPLER
000065            DO 10 N=1,50                                          KEPLER
000066            FOFF=SIN(OLDE)                                        KEPLER
000067            DFOFF=COS(OLDE)                                       KEPLER
000068            E=OLDE+(M-OLDF+ECC*FOFE)/(1.-ECC*DFOFF)               KEPLER
000069            IF(ABS((OLDF-F)/OLDE).LE.1.0E-07) GO TO 7             KEPLER
000070            OLDE=E                                                KEPLER
000071         10 CONTINUE                                              KEPLER
000072            ERROR=1.0                                             KEPLER
000073          7 TEMP = E                                              KEPLER
000074            E = PIMOD(TEMP)                                       KEPLER
000075          8 XE=E                                                  KEPLER
000076            RETURN                                                KEPLER
000077            END                                                   KEPLER
```

```
* ILT LATSEL,1,750922, 69845   , 1

000001          SUBROUTINE LATSEL                                          LATSEL
000002    C     THIS ROUTINE RESTORES SWATH DATA IN I/O AREA AND COMPUTES MIN-MAX LATSEL
000003    C     DELTA LONGITUDE TABLES                                     LATSEL
000004          COMMON /SWAT/                                              SWAT
000005         1DLONMN(100),DLONMX(100),TIME1(100,3),DLONG1(100,3),TNLAT(2),  SWAT
000006         2TNLAT(2),ISOSTR,NLAT,LATNO(100),IALT(100),ICRDI(17)        SWAT
000007          COMMON /REVSTH/                                            REVSTH
000008         1TTNE(181,3,2),DLONG(181,3,2),ALT(181,2)                    REVSTH
000009          COMMON /CONS/                                              CONS
000010         1ECCF,C15,CNUV,RADIUS,HEADER(20),NAME(2),NAME1(2),TWOPI,ALPHAM,  CONS
000011         2TMES,W,ORBITI(6),ORBIT2(6),PI,CJ2,CMU,PI2,PI32,SA(5),RADIAN,  CONS
000012         3ICASE,IPPIS(10),NLINE,NPAGE,INP,LINMAX,ISKIP,ISKR,NVEH,NODAY,  CONS
000013         4IV1TIM(6),IV2TIM(6),KO,TORBIT,LIDEBG(4)                    CONS
000014          REAL NAME,NAME1                                            CONS
000015          LOGICAL LIDEBG                                             CONS
000016          COMMON /ACCNUM/ ACCNUM(100)                                LATSEL
000017          EQUIVALENCE(ACCNUM(1),KVEH)                                LATSEL
000018          DO 50 I=1,NLAT                                             LATSEL
000019          IF(LATNO(I) .GE. 0)GO TO 5                                 LATSEL
000020          ISUB = IABS(LATNO(I))+91                                   LATSEL
000021          GO TO 10                                                   LATSEL
000022        5 ISUB =91 - LATNO(I)                                        SET2
000023       10 IALT(I) = (ALT(ISUB,KVEH))*10.*RADIUS                      SET2
000024          DO 15 K=1,3                                                LATSEL
000025          TIME1(I,K) = TIME(ISUB,K,KVEH)                             LATSEL
000026          DLONG1(I,K) = PIMOD(DLONG(ISUB,K,KVEH))                    SET2
000027       15 CONTINUE                                                   LATSEL
000028          OMEG1 = TWOPI + DLONG1(I,1) - 2*DLONG1(I,2)                LATSEL
000029          OMEG2 = TWOPI + DLONG1(I,3) - 2*DLONG1(I,2)                LATSEL
000030          IF(OMEG1 .GE. OMEG2)GO TO 20                               LATSEL
000031          IF(OMEG2 .GT. DLONMX(I))DLONMX(I) = OMEG2                  LATSEL
000032          IF(OMEG1 .LT. DLONMN(I))DLONMN(I) = OMEG1                  LATSEL
000033          GO TO 40                                                   LATSEL
000034       20 IF(OMEG1 .GT. DLONMX(I))DLONMX(I) = OMEG1                  LATSEL
000035          IF(OMEG2 .LT. DLONMN(I))DLONMN(I) = OMEG2                  LATSEL
000036       50 CONTINUE                                                   LATSEL
000037          RETURN                                                     LATSEL
000038          END                                                        LATSEL
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
* FCN LFPA,1,750925, 58785      , 1

000001          SUBROUTINE LFPA(FLDA,LMO,LYR,ALFGM,DAYS)                          LFPA
000002    C                                                                       LFPA
000003    C                                                                       LFPA     **-7
000004    C  GIVEN  DAY,MONTH,YEAR - SUBR. LFPA RETURNS THE RIGHT ASCENSION OF    LFPA
000005    C  GREENWICH AT MIDNIGHT OF A GIVEN DAY                                 LFPA
000006    C                                                                       LFPA
000007          COMMON /CONS/                                                     CONS
000008         1ECCE,CIS,CNUV,RADIUS,HEADER(20),NAME(2),NAME1(2),TWOPI,ALPHAM,    CONS
000009         2TIMES,WE,URBTT1(6),URBTT2(6),PI,CJ2,CMU,PI2,PI32,SA(3),RADIAN,    CONS
000010         3ICASE,IPP15(10),NLINE,NPAGE,INP,LINMAX,ISMIN,ISMR,NVEH,NUDAY,     CONS
000011         4IV1TIM(6),IV2TIM(6),KO,IURBIT,LIDEBG(4)                           CONS
000012          REAL NAME,NAME1                                                   CONS
000013          LOGICAL LIDEBG                                                    CONS
000014    C     REFERENCE EPOCH IS 0 HOUR 1JAN1950                                LFPA
000015    C                                                                       LFPA
000016    C     COMPUTE DAYS IN FULL YEARS FROM EPOCH TO LYR                      LFPA
000017    C     DAYS TO 1JAN1963 IS 4748.                                         LFPA
000018    C                                                                       LFPA
000019          DAYS=4748.                                                        LFPA
000020    C     NOTE--LYR MUST BE GREATER THAN OR EQUAL TO (19)64                 LFPA
000021          LASTYR=LYR-1                                                      LFPA
000022          DO 20 I=63,LASTYR                                                 LFPA
000023          KREMD=MOD((I-60),4)                                               LFPA
000024          IF(KREMD.GT.0) GO TO 10                                           LFPA
000025          DAYS=DAYS+366                                                     LFPA
000026          GO TO 20                                                          LFPA
000027       10 DAYS=DAYS+365                                                     LFPA
000028       20 CONTINUE                                                          LFPA
000029          IF(LMO-1)30,40,30                                                 LFPA
000030       30 IF(LMO-2)60,50,60                                                 LFPA
000031       40 DAYS=DAYS+FLDA-1.                                                 LFPA
000032          GO TO 270                                                         LFPA
000033       50 DAYS=DAYS+FLDA+30.                                                LFPA
000034          GO TO 270                                                         LFPA
000035       60 KDEL=MOD((LYR-60),4)                                              LFPA
000036          IF(KDEL.GT.0) GO TO 70                                            LFPA
000037          DAYS=DAYS+59.                                                     LFPA
000038          GO TO 80                                                          LFPA
000039       70 DAYS=DAYS+58.                                                     LFPA
000040       80 IF(LMO-3) 270,170,90                                              LFPA
000041       90 IF(LMO-5) 180,190,100                                             LFPA
000042      100 IF(LMO-7) 200,210,110                                             LFPA
000043      110 IF(LMO-9) 220,230,120                                             LFPA
000044      120 IF(LMO-11)240,250,260                                             LFPA
000045      170 DAYS=DAYS+FLDA                                                    LFPA
000046          GO TO 270                                                         LFPA
000047      180 DAYS=DAYS+FLDA+31.                                                LFPA
000048          GO TO 270                                                         LFPA
000049      190 DAYS=DAYS+FLDA+61.                                                LFPA
000050          GO TO 270                                                         LFPA
000051      200 DAYS=DAYS+FLDA+92.                                                LFPA
000052          GO TO 270                                                         LFPA
000053      210 DAYS=DAYS+FLDA+122.                                               LFPA
000054          GO TO 270                                                         LFPA
000055      220 DAYS=DAYS+FLDA+153.                                               LFPA
000056          GO TO 270                                                         LFPA
000057      230 DAYS=DAYS+FLDA+184.                                               LFPA
000058          GO TO 270                                                         LFPA
```

```
000059          240 DAYS=DAYS+FLDA+214.                                   LFPA
000060              GO TO 270                                             LFPA
000061          250 DAYS=DAYS+FLDA+245.                                   LFPA
000062              GO TO 270                                             LFPA
000063          260 DAYS=DAYS+FLDA+275.                                   LFPA
000064          270 CONTINUE                                              LFPA
000065              ALEGM=AMOD(SNGL((.100075542D3+(.985647346D0)*DAYS+    LFPA
000066             1(2.90150-13)*DAYS**2))+360.)                          LFPA
000067              ALEGM=ALEGM/RADIAN                                    LFPA
000068              RETURN                                                LFPA
000069              END                                                   LFPA
```

```
• FLT MAIN•1•760331• 59925      • 1

000001          C     PROGRAM MAIN(ITEST=401•OTEST=401•TAPE5=ITEST•TAPE6=OTEST•          SET7
000002          C    1SWATH=1500•SWATHR=500•TAPE8=SWATH•TAPE9=SWATHR)                    SET7
000003          C     THIS IS THE MAIN CONTROL ROUTINE FOR EPHEM PROGRAM                  MAIN
000004                COMMON /FLAG/                                                       FLAG
000005               1IFIRST•IVEH•ISTART•IEND•IDEND                                       FLAG
000006                COMMON /CONS/                                                       CONS
000007               1ECCE•CIS•CNUV•RADIUS•HEADER(20)•NAME(2)•NAME1(2)•TWOPI•ALPHAM•      CONS
000008               2THES•WE•ORBIT1(6)•ORBIT2(6)•PI•CJ2•CHU•PI2•PI32•SA(3)•RADIAN•       CONS
000009               3ILAT•IPRIS(10)•NLINL•NPAGE•INP•LINMAX•ISWTH•ISWR•NVEH•NODAY•        CONS
000010               4IV1TIM(6)•IV2TIM(6)•KD•TORBIT•LIDEBG(4)                             CONS
000011                REAL NAME•NAME1                                                     CONS
000012                LOGICAL LIDEBG                                                      CONS
000013                COMMON /HECCOM/                                                     HECCOM
000014               1ALPHAT•XBETAD(2)•XBETA(2•2)•XND(2)•XOMEGD(2)•TASCNP(2)•XT0(2)•      HECCOM
000015               2PERIOD(2)•LEO•LDA•NREV1•NREV2•IYR•IYRP•IDAY•IPDAY•ICDAY•NREV1P•     HECCOM
000016               3NREV2P•IV1DAY•IV2DAY•LYR                                            SET1
000017                COMMON /ACCEPH/                                                     ACCEPH
000018               1PNODE(17•2)•SK(1•2)•EC(1•2)•XIN(1•2)•RTASC(1•2)•WP(1•2)•TNDE(1•2)   ACCEPH
000019               2•RG(1)                                                              ACCEPH
000020                COMMON /NREV/                                                       NREV
000021               1XLONG(1•2)•NREV(2)                                                  NREV
000022                DIMENSION NREVCT(1•2)                                               NREV
000023                EQUIVALENCE (NREV(1)•NREVCT(1•1))                                   NREV
000024                COMMON /SWAT/                                                       SWAT
000025               1DLONGN(100)  DLONGX(100)•TIME1(100•3)•DLONG1(100•3)•INLAT(2)•       SWAT
000026               2ISLAT(2)•ISTR•NEAT•LATND(100)•IALT(100)•IGRDI(17)                   SWAT
000027                COMMON /ACCNUM/ ACCNUM(100)                                         MAIN
000028                EQUIVALENCE(ACCNUM(1)•NVEH)•(ACCNUM(89)•KDAY)                       MAIN
000029                EQUIVALENCE(IYEAR•IYR)                                              MAIN
000030                COMMON/ICARUS/ CARUS(8•17•2)                                        MAIN
000031                COMMON /XDAY/ IDAS(2)                                               SET7
000032                DATA CONS/10.E10/•CONS1/-10.E10/                                    MAIN
000033                IFIRST = 1                                                          MAIN
000034                DO 5 I=1•1101                                                       MAIN
000035              5 CONTINUE                                                            MAIN
000036                NREV1 = 1                                                           MAIN
000037                NREV2 = 1                                                           MAIN
000038                CALL INPROC                                                         MAIN
000039                DO 15 I=1•6                                                         MAIN
000040                CARUS(I•1•1) = ORBIT1(I)                                            MAIN
000041                CARUS(I•1•2) = ORBIT2(I)                                            MAIN
000042             15 CONTINUE                                                            MAIN
000043                FLDA = IV1TIM(3)                                                    DEBUG
000044                CALL TPA(FLDA•IV1TIM(2)•IV1TIM(1)•ZZ•DAYS)                          DEBUG
000045                IDAY1 = DAYS                                                        MAIN
000046                ISTART = IDAY1                                                      *NEW
000047                XHU = CHU                                                           *NEW
000048                IF(IVEH .LT. 2)GO TO 17                                             *NEW
000049                FLDA =IV2TIM(3)                                                     DEBUG
000050                CALL TPA(FLDA•IV2TIM(2)•IV2TIM(1)•ZZ•DAYS)                          DEBUG
000051                IDAY2 = DAYS                                                        MAIN
000052                IF(IDAY1 .LE. IDAY2)ISTART = IDAY1                                  MAIN
000053                IF(IDAY1 .GT. IDAY2)ISTART = IDAY2                                  MAIN
000054             17 IEND = ISTART + NODAY - 1                                           *NEW
000055                IDAS(1) = IDAY1                                                     SET7   **-1
000056                IDAS(2) = IDAY2                                                     SET7
000057                DO 200 J=1•LEN                                                      MAIN
000058                DO 10 I=1•100                                                       SET3
```

```
000059          DLONMX(I) = CONS1                                   SET3
000060          DLONMN(I) = CONS                                    SET3
000061       10 CONTINUE                                            SET3
000062          IF(I .EQ. 1)LYR = IV1TIM(1)                         DEBUG
000063          IF(J .EQ. 2)LYR =IV2TIM(1)                          DEBUG
000064          IDEND = NODAY                                       MAIN
000065          IF(J .GT. 1)GO TO 30                                MAIN
000066          IYEAR = IV1TIM(1)                                   MAIN
000067          IYEAR = MOD(IYEAR,4)                                MAIN
000068          ICDAY = 0                                           MAIN
000069          CALL DAY(IV1TIM,IV1DAY)                             MAIN
000070          IPDAY = IV1DAY - 2                                  MAIN
000071          IDAY = IV1DAY - 1                                   MAIN
000072          IYEARB = IV1TIM(1) + 1900 - 1                       MAIN
000073          IF(IDAY1 .GT. ISTART)IDEND = NODAY -(IDAY1 - ISTART)  MAIN
000074          GO TO 40                                            MAIN
000075       30 IYEAR = IV2TIM(1)                                   MAIN
000076          IYEAR = MOD(IYEAR,4)                                MAIN
000077          ICDAY = 0                                           MAIN
000078          CALL DAY(IV2TIM,IV2DAY)                             MAIN
000079          IPDAY = IV2DAY - 2                                  MAIN
000080          IDAY  = IV2DAY - 1                                  MAIN
000081          IYEARB = IV2TIM(1) + 1900 - 1                       MAIN
000082          IF(IDAY2 .GT. ISTART)IDEND = NODAY -(IDAY2 - ISTART)  MAIN
000083       40 IF(IDEND.GT. 0)GO TO 45                             MAIN
000084          CALL PAGER(2)                                       DEBUG
000085          WRITE(60,905)                                       DEBUG
000086          GO TO 500                                           DEBUG
000087       45 RHO = CARUS(1,1,J    )*(1. -(CARUS(2,1,J    ))**2)   DEBUG
000088          XG = 1. - 1.5*(SIN(CARUS(3,1,J    ))**2)
000089          YG = 1.5*COS*SQRT( 1. -(CARUS(2,1,J    ))**2)/RHO**2   *NEW
000090          XK = 1. + YG*XG                                     *NEW
000091          CMU = XMU*XK**2                                     *NEW
000092          DO 170 I=1,IDEND                                    *NEW
000093          ICDAY = ICDAY + 1                                   *NEW
000094          IDAY = IDAY + 1                                     *NEW
000095          IPDAY = IPDAY + 1                                   *NEW
000096          NREV1P = NREV1                                      MAIN
000097          NREV2P = NREV2                                      **-1
000098          IF(MOD(IYEARB+IYEAR,4) .EQ. 0)GO TO 50              MAIN
000099          IF(IDAY .LT. 366)GO TO 60                           MAIN
000100          GO TO 55                                            MAIN
000101       50 CONTINUE                                            MAIN
000102          IF(IDAY .LT. 367)GO TO 60                           MAIN
000103          IYEAR = MOD(IYEAR+1,4)                              MAIN
000104          IDAY = IDAY - 366                                   MAIN
000105          GO TO 60                                            MAIN
000106       55 CONTINUE                                            MAIN
000107          IYEAR = MOD(IYEAR+1,4)                              MAIN
000108          IDAY = IDAY - 365                                   MAIN
000109       60 CONTINUE                                            MAIN
000110          IF(J .GT. 1)NREV(J)   = NREV2                       DEBUG
000111          IF(J .EQ. 1)NREV(J)   = NREV1                       MAIN
000112          KDAY = I                                            SET9
000113          KVEH = J                                            MAIN
000114          CALL VECTOR                                         MAIN
000115          IF(KVEH .EQ. 1)NREV(KVEH) = NREV1                   DEBUG
000116          IF(KVEH .GT. 1)NREV(KVEH) = NREV2                   MAIN
000117          IF(IPRST .EQ. 0)GO TO 70                            SET2
000118          CALL WRITE6(1)                                      SET2
                                                                    MAIN
                                                                    MAIN
```

```
000119          IFIRST = 0                                                      MAIN
000120       70 SN(1,J) = CARUS(1,1,J)                                          MAIN
000121          EC(1,J) = CARUS(2,1,J)                                          MAIN
000122          XTN(1,J) = CARUS(3,1,J)                                         MAIN
000123          RTASC(1,J) = CARUS(8,1,J)                                       DEBUG
000124          WP(1,J) = PIMOD(CARUS(5,1,J))                                   DEBUG
000125          TNDE(1,J) = TASCNP(J)                                           DEBUG
000126          RG(1) = PIMOD(CARUS(6,1,J) - CARUS(4,1,J) - WE*TASCNP(J))       DEBUG
000127       80 CONTINUE                                                        MAIN
000128          CALL ACCGEN                                                     MAIN
000129          DO 90 L=1,10,2                                                  MAIN
000130          IF(IPPTS(L) .LE. KDAY .AND. IPPTS(L+1) .GE. KDAY)GO TO 100      MAIN
000131       90 CONTINUE                                                        MAIN
000132          GO TO 170                                                       MAIN
000133      100 CALL PRTIS                                                      MAIN
000134      170 CONTINUE                                                        MAIN
000135          CALL WRITSW(3)                                                  MAIN
000136      200 CONTINUE                                                        MAIN
000137          CALL WRITSW(4)                                                  MAIN
000138          CALL EJECT(4)                                                   MAIN
000139          WRITE(KO,900)                                                   MAIN
000140      500 STOP                                                            DEBUG
000141      905 FORMAT(1H0,68H*** THE DIFFERENCE BETWEEN VEHICLE START DATES IS GREDEBUG
000142        TEATER THAN NODAY)                                                DEBUG
000143      900 FORMAT(1H0,15X,39H*** SWATH GENERATION HAS BEEN COMPLETED)      SETP
000144          END                                                             MAIN
```

```
@ ELT PAGER,1,760331, 59915     , 1
100001          SUBROUTINE PAGER(NLINES)                                        PAGER
000002    C                                                                     PAGER
000003    C                                                                     PAGER
000004          COMMON /CONS/                                                   CONS
000005         1ECCE,CIS,CMUV,RADIUS,HEADER(20),NAME(2),NAME1(2),TWOPI,ALPHAM,  CONS
000006         2TMES,PE,ORBIT1(6),ORBIT2(6),PI,CJ2,CMU,PI2,PI32,SA(3),RADIAN,   CONS
000007         3ICASE,IPPTS(10),NLINE,NPAGE,INP,LINMAX,ISWTR,ISNK,NVEH,NODAY,   CONS
000008         4IV1TIM(6),IV2TIM(6),KO,TORBIT,LIDEBG(4)                         CONS
000009          REAL NAME,NAME1                                                 CONS
000010          LOGICAL LIDEBG                                                  PAGER
000011          NLINE=NLINE+NLINES                                              PAGER
000012          IF (NLINE.LT.LINMAX) GO TO 300                                  PAGER
000013          IF (NPAGE.EQ.0) GO TO 200                                       PAGER
000014          LMAX=LINMAX-NLINE+NLINES                                        PAGER
000015          GO TO 10                                                        PAGER
000016       10 CONTINUE                                                        PAGER
000017    C                                                                     PAGER
000018    C  SPACE TO BOTTOM OF PAGE                                            PAGER
000019    C                                                                     PAGER
000020          IF (NLINES.LT.0) RETURN                                         PAGER
000021      200 CONTINUE                                                        PAGER
000022          NLINE=NLINES                                                    PAGER
000023          NPAGE=NPAGE+1                                                   PAGER
000024          WRITE (KO,9010)                                                 PAGER
000025          WRITE(KO,9020)(HEADER(I),I=1,18),ICASE,NPAGE                    PAGER  *NEW
000026      300 CONTINUE                                                        PAGER  **-1
000027          RETURN                                                          PAGER
000028    C                                                                     PAGER
000029    C  FORMATS                                                            PAGER
000030    C                                                                     PAGER
000031     9000 FORMAT (1H )                                                    PAGER
000032     9010 FORMAT (1H1)                                                    PAGER
000033     9020 FORMAT(1X,18A4,2X,16HLPP SIMULATION  ,5HCASE ,I4,2X,4HPAGE,I4)  PAGER  *NEW
000034    C                                                                     PAGER  **-1
000035          END                                                             PAGER
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
* ELT PIMOD,1,750925, $8784     , 1

      FUNCTION PIMOD(A)                                                 PIMOD
C                                                                       PIMOD
C                                                                       PIMOD    **-5
C  GET POSITIVE ARGUMENT OF AN ANGLE IN RADIANS BETWEEN 0 AND 2PI       PIMOD
C                                                                       PIMOD
      B=6.2831853                                                       PIMOD
   10 IF(A) 20,50,30                                                    PIMOD
   20 A=A+B                                                             PIMOD
      GO TO 10                                                          PIMOD
   30 IF(A-B) 50,50,40                                                  PIMOD
   40 A=A-B                                                             PIMOD
      GO TO 30                                                          PIMOD
   50 PIMOD=A                                                           PIMOD
      RETURN                                                            PIMOD
      END                                                               PIMOD
```

```
* FLT PRT15,1,750922, 69855     , 1

000001          SUBROUTINE PRT15                                          PRT15
000002    C     THIS ROUTINE PRINTS EPHEMERIS DATA FOR 1 DAY AND 1 VEHICLE PRT15
000003          COMMON /HECCOM/                                           HECCOM
000004         1ALPHAT,XBETAD(2),XBETA(2,2),XWD(2),XOMEGD(2),TASCNP(2),XT0(2),  HECCOM
000005         2PERTOD(2),LMO,LDA,NREV1,NREV2,IYR,IYRP,IDAY,IPDAY,ICDAY,NREV1P, HECCOM
000006         3NREV2P,IV1DAY,IV2DAY,LYP                                 SET1
000007          COMMON /CONS/                                             CONS
000008         1ECCF,CIS,CNUV,RADIUS,HEADER(20),NAME(2),NAME1(2),TWOPI,ALPHAH,  CONS
000009         2THES,WT,URBIT1(6),URBIT2(6),PI,CJ2,CHU,P12,P132,SA(3),RADIAN,  CONS
000010         3ICASE,IPP15(10),NLINE,NPAGE,INP,LIMMAX,ISWTB,ISWR,NVEH,NODAY,  CONS
000011         4IV1TIM(6),IV2TIM(6),KO,TORBIT,L10FDG(4)                  CONS
000012          REAL NAME,NAME1                                           CONS
000013          LOGICAL L10EBG                                            CONS
000014          COMMON /FLAG/                                             FLAG
000015         1IFIRST,IVEH,ISTART,IEND,IDEND                             FLAG
000016          COMMON /ACCEPH/                                           ACCEPH
000017         1TNODE(17,2),SM(1,2),EC(1,2),XIN(1,2),RTASC(1,2),WP(1,2),TNDE(1,2)ACCEPH
000018         2,RG(1)                                                    ACCEPH
000019          COMMON /NREV/                                             NREV
000020         1XLONG(1,2),NREV(2)                                        NREV
000021          DIMENSION NREVCT(1,2)                                     NREV
000022          EQUIVALENCE (NREV(1),NREVCT(1,1))                         NREV
000023          COMMON/1CARUS/ CARUS(8,17,2)                              PRT15
000024          COMMON/ACCNUM/ ACCNUM(100)                                PRT15
000025          EQUIVALENCE(ACCNUM(7),PER(1)),(ACCNUM(85),DELNDE(1)),(ACCNUM(89), PRT15
000026         1KDAY),(ACCNUM(1),KVEH)                                    SET1
000027          REAL PER(2),DELNDE(2)                                     PRT15
000028          INTEGER XDEG(3,2),WDEG(3,2),RDEG(3,2),DEG(3)              PRT15
000029          JDAY = 1                                                  PRT15
000030          CALL EJECT(14)                                            PRT15
000031          WRITE(KO,100)LMO,LDA,LYR,KVEH                             SET4
000032      100 FORMAT(/33X,42HT-5  PREDICTOR EPHEMERIS DISPLAY FOR DATE ,I2,1H/, SET3
000033         1I2,1H/,I2,///,44X                                         SET3
000034         1,4HVEHICLE ,I1)                                           PRT15
000035          SMNT1 = SM(JDAY,KVEH)*RADIUS                              PRT15
000036          CALL DEGMOD(XIN(JDAY,KVEH),XDEG(1,KVEH))                  PRT15
000037          CALL DEGMOD(WP(JDAY,KVEH),WDEG(1,KVEH))                   PRT15
000038          CALL DEGMOD(RG(JDAY),RDEG(1,KVEH))                        PRT15
000039          ECCPR = EC(JDAY,KVEH)                                     SET2
000040          PERPR = PER(KVEH)                                         SET2
000041          WRITE(KO,150)SMNT1,ECCPR,PERPR,(XDEG(K,KVEH),K=1,3),(WDEG(K,KVEH),PRT15
000042         1K=1,3),(RDEG(K,KVEH),K=1,3)                               PRT15
000043      150 FORMAT(/37X,21HSEMI-MAJOR AXIS      =, F10.3,             PRT15
000044         1/37X,21HECCENTRICITY          =,F11.8,                    PRT15
000045         2/37X,21HPERIGEE               =,F10.3,                    PRT15
000046         3/37X,21HINCLINATION           =,I4,2I3,                   PRT15
000047         4/37X,21HARGUEMENT OF PERIGEE=,I4,2I3,                     PRT15
000048         5/37X,21HRT. ASC. GREENWICH  =,I4,2I3)                     PRT15
000049          WRITE(KO,169)                                             PRT15
000050      169 FORMAT(/37X,31HREV   NODE TIME  NODE LONGITUDE)           PRT15
000051          MIN = NREV(KVEH)                                          PRT15
000052          DO 210 J=1,MIN                                            PRT15
000053          CALL DEGMOD(CARUS(4,J,KVEH),XDEG(1,KVEH))                 PRT15
000054          CALL PAGER(1)                                             PRT15
000055          WRITE(KO,190)(J,TNODE(J,KVEH),(XDEG(K,KVEH),K=1,3))       PRT15
000056      190 FORMAT(37X,I2,1H)F12.4,3X,I4,2I3)                         PRT15
000057      210 CONTINUE                                                  PRT15
000058          RETURN                                                    PRT15
```

END

```
* FLT REVTAB,1,750925, 58796   , 1

000001          SUBROUTINE REVTAB                                       REVTAB
000002     C                                                            REVTAB
000003     C                                                            REVTAB  *-11
000004     C                                                            REVTAB
000005     C  SUBROUTINE REVTAB RECEIVES AS INPUT THE SWATH DATA TABLE FROM THE   REVTAB
000006     C  SUBROUTINE SWATH. REVTAB REFORMATS THE TABLE TO OBTAIN THE DATA AT  REVTAB
000007     C  EACH INTEGER VALUE OF LATITUDE.                              REVTAB
000008     C                                                            REVTAB
000009     C                                                            REVTAB
000010          COMMON /CONS/                                           CONS
000011         1ECCF,CIS,CNUV,RADIUS,HEADEN(20),NAME(2),NAME1(2),TWOPI,ALPHAM,  CONS
000012         2THES,WE,ORBIT1(6),ORBIT2(6),PI,CJ2,CMU,PI2,PI32,SA(3),RADIAN,  CONS
000013         3ICASE,IPPTS(10),NLINE,NPAGE,INP,LINMAX,ISWTH,ISWR,NVEH,NUDAY,  CONS
000014         4IV1TIM(6),IV2TIM(6),KO,TORBIT,LIDEBG(4)                CONS
000015          REAL NAME,NAME1                                         CONS
000016          LOGICAL LIDEBG                                          CONS
000017          DIMENSION NO(5) ,XLONND(19,2)                           REVTAB
000018          DIMENSION MAXLT(2),MINLT(2)                             REVTAB
000019          COMMON/ACCNUM/ ACCNUM(100)                              REVTAB
000020          EQUIVALENCE (ACCNUM(  1),KVEH)                          REVTAB
000021         X,           (ACCNUM(  3),NUSCN)                         REVTAB
000022         X,           (ACCNUM(  4),NADSCN)                        REVTAB
000023         X,           (ACCNUM( 87),XLONND(1,1))                   REVTAB
000024         X,           (ACCNUM( 89),KUAY)                          REVTAB
000025         X,           (ACCNUM( 90),F)                             REVTAB
000026         X,           (ACCNUM( 92),MAXLT(1))                      REVTAB
000027         X,           (ACCNUM( 94),MINLT(1))                      REVTAB
000028         X,           (ACCNUM(  96),IADF)                         REVTAB
000029     C                                                            REVTAB
000030     C                                                            REVTAB
000031          COMMON /ACCDAT/                                         ACCDAT
000032         1TV(362),HV(362),SLAM(362,3),SPHI(362,3),ENP(2)          ACCDAT
000033          COMMON /REVSTH/                                         REVSTH
000034         1TIME(181,3,2),DLONG(181,3,2),ALT(181,2)                 REVSTH
000035          COMMON /ACCEPH/                                         ACCEPH
000036         1TMNODE(1,2),SH(1,2),EC(1,2),XIN(1,2),RTASC(1,2),WP(1,2),TNDE(1,2)ACCEPH
000037         2,RG(1)                                                  ACCEPH
000038          COMMON /FLAG/                                           FLAG
000039         1IFIRST,IVEH,ISTART,IEND,IDEND                           FLAG
000040     C                                                            REVTAB
000041     C                                                            REVTAB
000042     C                                                            REVTAB
000043     C  INITIALIZE OUTPUT ARRAYS FOR THIS VEHICLE                 REVTAB
000044     C                                                            REVTAB
000045          DO 10   I = 1,181                                       REVTAB
000046     C                                                            REVTAB
000047          ALT(I,KVEH) = 0.0                                       REVTAB
000048     C                                                            REVTAB
000049          DO 10   K = 1,3                                         REVTAB
000050     C                                                            REVTAB
000051          DLONG(I,K,KVEH) = 2.E6                                  REVTAB
000052       10 TIME(I,K,KVEH) = 2.E6                                   REVTAB
000053     C                                                            REVTAB
000054     C  DETERMINE ENTRY POINT IN SWATH DATA THAT CORRESPONDS TO N.P.  REVTAB
000055     C                                                            REVTAB
000056          TEMP = ENP(KVEH) - F                                    REVTAB
000057          IF (TEMP .LT. 0.0) TEMP = TEMP + TWOPI                  REVTAB
000058          NTRY = TEMP/RADIAN                                      REVTAB
```

```
000059          C                                                               REVTAB
000060           20   IF (TV(NTRY) .GT. TV(362))  GO TO 50                      REVTAB
000061                NTRY = NTRY + 1                                           REVTAB
000062                GO TO 20                                                  REVTAB
000063          C                                                               REVTAB
000064           50   IF (TV(NTRY) .LE. TV(362)) GO TO 100                      REVTAB
000065                NTRY = NTRY - 1                                           REVTAB
000066                GO TO 50                                                  REVTAB
000067          C                                                               REVTAB
000068          C  INITIALIZE FOR DESCENDING PASS SWATH DATA                    REVTAB
000069          C                                                               REVTAB
000070          100 CONTINUE                                                    REVTAB
000071                TMXLAT = 0.                                               REVTAB
000072                XMXLAT = 0.                                               REVTAB
000073                TMNMX = 1.E6                                              REVTAB
000074                TMXMN = -100.                                             REVTAB
000075                TMNLAT = 0.                                               REVTAB
000076                XMNLAT = 0.                                               REVTAB
000077                JJ = 1                                                    REVTAB
000078          C  DETERMINE LATITUDE EXTREMES OF ACCESS TABLES                 REVTAB
000079          C                                                               REVTAB
000080                DO 105 M = 1,NOSCN                                        REVTAB
000081                TEMP = SLAM(362,M)                                        REVTAB
000082                IF (TEMP .GT. 90.) GO TO 105                              REVTAB
000083                TMNMX = AMIN1(TMNMX,TEMP)                                 REVTAB
000084                TMXLAT = AMAX1(TMXLAT,TEMP)                               REVTAB
000085                IF (TEMP .GE. TMXLAT) GO TO 105                           REVTAB
000086                XMXLAT = AMAX1(XMXLAT,TEMP)                               REVTAB
000087          105 CONTINUE                                                    REVTAB
000088          C                                                               REVTAB
000089          110 CONTINUE                                                    REVTAB
000090                MXLAT = 91 - TMXLAT                                       REVTAB
000091                MNMX = 91 - TMNMX                                         REVTAB
000092          C                                                               REVTAB
000093                DO 115 M = 1,NOSCN                                        REVTAB
000094                TEMP = SLAM(361,M)                                        REVTAB
000095                IF (TEMP .GT. 90.) GO TO 115                              REVTAB
000096                TMXMN = AMAX1(TMXMN,TEMP)                                 REVTAB
000097                TMNLAT = AMIN1(TMNLAT,TEMP)                               REVTAB
000098                IF (TEMP .LE. TMNLAT) GO TO 115                           REVTAB
000099                XMNLAT = AMIN1(XMNLAT,TEMP)                               REVTAB
000100          115 CONTINUE                                                    REVTAB
000101          C                                                               REVTAB
000102                MXMN = 91 - TMXMN                                         REVTAB
000103                MNLAT = 91 - TMNLAT                                       REVTAB
000104          C                                                               REVTAB
000105          C                                                               REVTAB
000106          C  INITIALIZE POINTERS TO SWATH TABLE FOR EACH SCAN ANGLE       REVTAB
000107          C                                                               REVTAB
000108          125  DO 150  I = 1,3                                            REVTAB
000109          150 NO(I) = NTRY                                                REVTAB
000110                IF(.NOT. LDEBG(4)) GO TO 160                              REVTAB
000111                CALL PAGER(7)                                             SET2
000112                WRITE(KO,9030) MXLAT,MNMX,MXMN,MNLAT                      REVTAB
000113                WRITE(KO,9040) (SLAM(361,K1),K1=1,NOSCN)                  REVTAB
000114                WRITE(KO,9050) (SLAM(362,K1),K1=1,NOSCN)                  REVTAB
000115                WRITE(KO,9025)                                            REVTAB
000116          C                                                               REVTAB
000117          C  COMPUTE SCAN DATA FOR EACH LAT. ABOVE MINIMUM OF SWATH EDGES AT N.P. REVTAB
000118          C                                                               REVTAB
```

```
000119   160  CONTINUE                                                    REVTAB
000120  C                                                                 REVTAB
000121       DO 300  I  = MXLAT+MNMX                                      REVTAB
000122  C                                                                 REVTAB
000123       XI  = 91 - I                                                 REVTAB
000124       INT = I                                                      REVTAB
000125       DO 250  K = 1+NOSCN                                          REVTAB
000126  C                                                                 REVTAB
000127       IF (SLAM(362,K) .LT. XI) GO TO 250                           REVTAB
000128   175 L = NO(K)                                                    REVTAB
000129       IF (SLAM(L,K) .LE. XI) GO TO 200                             REVTAB
000130       NO(K) = NO(K) + JJ                                           REVTAB
000131       GO TO 175                                                    REVTAB
000132  C                                                                 REVTAB
000133  C  INTERPOLATE BETWEEN TWO VALUES FOR THIS SCAN ANGLE FROM TABLE TO REVTAB
000134  C  OBTAIN DATA FOR INTEGER LATITUDE                               REVTAB
000135  C                                                                 REVTAB
000136   200 IF( SLAM(L-JJ,K) .LE. 1.E6 .AND. SLAM(L,K) .NE. 1.E6) GO TO 210 REVTAB
000137       TIME(INT,K,KVEH) = 1.E6                                      REVTAB
000138       DLONG(INT,K,KVEH) = 1.E6                                     REVTAB
000139       GO TO 250                                                    REVTAB
000140   210 DEL = (XI-SLAM(L-JJ,K))/(SLAM(L,K)-SLAM(L-JJ,K))             REVTAB
000141       TIME(INT,K,KVEH) = (TV(L)-TV(L-JJ))*DEL + TV(L-JJ)           REVTAB
000142       DLONG(INT,K,KVEH) = (SPHI(L,K)-SPHI(L-JJ,K))*DEL + SPHI(L-JJ,K) REVTAB
000143   250 CONTINUE                                                     REVTAB
000144  C                                                                 REVTAB
000145       K = NADSCN                                                   REVTAB
000146       L = NO(NADSCN)                                               REVTAB
000147       ALT(INT,KVEH) = ((XI-SLAM(L-JJ,K))/(SL    ,K)-SLAM(L-JJ,K)))  REVTAB
000148      *                    * (HV(L)-HV(L-JJ)) + HV(L-JJ)            REVTAB
000149   300 CONTINUE                                                     REVTAB
000150  C                                                                 REVTAB
000151  C                                                                 REVTAB
000152  C  PROCESS INTEGER LATITUDES FROM LOWEST SWATH EDGE AT N.P. TO HIGHEST REVTAB
000153  C  SWATH EDGE AT S.P.                                             REVTAB
000154  C                                                                 REVTAB
000155  C                                                                 REVTAB
000156   310 CONTINUE                                                     REVTAB
000157       I1 = MNMX + 1                                                REVTAB
000158       I2 = MXMN - 1                                                REVTAB
000159  C                                                                 REVTAB
000160       DO 400 INT = I1,MXMN                                         REVTAB
000161  C                                                                 REVTAB
000162  C                                                                 REVTAB
000163       XI  = 91 - INT                                               REVTAB
000164  C                                                                 REVTAB
000165       DO 375  K = 1+NOSCN                                          REVTAB
000166  C                                                                 REVTAB
000167       IFL = 0                                                      REVTAB
000168   325 L = NO(K)                                                    REVTAB
000169       IF (SLAM(L,K) .LE. XI) GO TO 350                             REVTAB
000170       NO(K) = NO(K) + JJ                                           REVTAB
000171       IF ( NO(K) .GT. 0 .AND. NO(K) .LT. 361) GO TO 325            REVTAB
000172       IF ( NO(K) .GT. 360) NO(K) = 1                               REVTAB
000173       IF( NO(K) .LE. 0) NO(K) = 360                                REVTAB
000174       IF ( IFL .GT. 0) GO TO 330                                   REVTAB
000175       IFL = 1                                                      REVTAB
000176       GO TO 325                                                    REVTAB
000177  C                                                                 REVTAB
000178   330 CONTINUE                                                     REVTAB
```

```
      TIME(INT,K,KVEH) = 1.E6                                           REVTAB
      DLONG(INT,K,KVEH) = 1.E6                                          REVTAB
      GO TO 375                                                         REVTAB
C                                                                       REVTAB
C  INTERPOLATE BETWEEN TWO VALUES FOR THIS SCAN ANGLE FROM TABLE TO     REVTAB
C  OBTAIN DATA FOR INTEGER LATITUDE                                     REVTAB
C                                                                       REVTAB
  350 CONTINUE                                                          REVTAB
      AA = SLAM(L,K) - SLAM(L-JJ,K)                                     REVTAB
      IF( ABS(AA) .GT. 0.0) GO TO 360                                   REVTAB
      LLL = NO(NADSCN)                                                  REVTAB
      CALL PAGER(2)                                                     REVTAB
      WRITE(KO,9060)                                                    REVTAB
      WRITE(KO,9070)K,L,AA,JJ,DEL,XI,INT,NTRY,LLL                       REVTAB
  360 IF( SLAM(L-JJ,K) .NE. 1.E6 .AND. SLAM(L,K) .NE. 1.E6) GO TO 365   REVTAB
      TIME(INT,K,KVEH) = 1.E6                                           REVTAB
      DLONG(INT,K,KVEH) = 1.E6                                          REVTAB
      GO TO 375                                                         REVTAB
  365 DEL = (XI-SLAM(L-JJ,K))/(SLAM(L,K)-SLAM(L-JJ,K))                  REVTAB
      TIME(INT,K,KVEH) = (TV(L)-TV(L-JJ))*DEL + TV(L-JJ)                REVTAB
      DLONG(INT,K,KVEH) = (SPHI(L,K)-SPHI(L-JJ,K))*DEL + SPHI(L-JJ,K)   REVTAB
  375 CONTINUE                                                          REVTAB
C                                                                       REVTAB
      K = NADSCN                                                        REVTAB
      L = NO(NADSCN)                                                    REVTAB
      AA = SLAM(L,K) - SLAM(L-JJ,K)                                     REVTAB
      IF ( ABS(AA) .GT. 0.0) GO TO 380                                  REVTAB
      LLL = NO(NADSCN)                                                  REVTAB
      CALL PAGER(2)                                                     REVTAB
      WRITE(KO,9060)                                                    REVTAB
      WRITE(KO,9070)K,L,AA,JJ,DEL,XI,INT,NTRY,LLL                       REVTAB
  380 CONTINUE                                                          REVTAB
      ALT(INT,KVEH) = ((XI-SLAM(L-JJ,K))/(SLAM(L,K)-SLAM(L-JJ,K)))      REVTAB
     *                 * (HV(L)-HV(L-JJ)) + HV(L-JJ)                    REVTAB
  400 CONTINUE                                                          REVTAB
  406 CONTINUE                                                          REVTAB
C                                                                       REVTAB
C                                                                       REVTAB
C  PROCESS INTEGER LATITUDES FROM HIGHEST SWATH EDGE AT S.P. TO LOWEST  REVTAB
C  SWATH EDGE AT S.P.                                                   REVTAB
C                                                                       REVTAB
C                                                                       REVTAB
C                                                                       REVTAB
C                                                                       REVTAB
      DO 500 INT = 12,HNLAT                                             REVTAB
C                                                                       REVTAB
C                                                                       REVTAB
      XI  =  91 - INT                                                   REVTAB
C                                                                       REVTAB
      DO 475  K = 1,NOSCN                                               REVTAB
C                                                                       REVTAB
      IF ( SLAM(361,K) .GT. XI) GO TO 475                               REVTAB
  425 L = NO(K)                                                         REVTAB
      IF (SLAM(L,K) .LE. XI) GO TO 450                                  REVTAB
      NO(K) = NO(K) + JJ                                                REVTAB
      GO TO 425                                                         REVTAB
C                                                                       REVTAB
C  INTERPOLATE BETWEEN TWO VALUES FOR THIS SCAN ANGLE FROM TABLE TO     REVTAB
C  OBTAIN DATA FOR INTEGER LATITUDE                                     REVTAB
C                                                                       REVTAB
```

```
  450 IF( SLAM(L-JJ,K) .NE. 1.E6 .AND. SLAM(L,K) .NE. 1.E6) GO TO 465   REVTAB
      TIME(INT,K,KVEH) = 1.E6                                           REVTAB
      DLONG(INT,K,KVEH) = 1.E6                                          REVTAB
      GO TO 475                                                         REVTAB
  465 DEL = (XI-SLAM(L-JJ,K))/(SLAM(L,K)-SLAM(L-JJ,K))                  REVTAB
      TIME(INT,K,KVEH) = (TV(L)-TV(L-JJ))*DEL + TV(L-JJ)                REVTAB
      DLONG(INT,K,KVEH) = (SPHI(L,K)-SPHI(L-JJ,K))*DEL + SPHI(L-JJ,K)   REVTAB
  475 CONTINUE                                                          REVTAB
C                                                                       REVTAB
      K = NADSCN                                                        REVTAB
      L = NO(NADSCN)                                                    REVTAB
      ALT(INT,KVEH) = ((XI-SLAM(L-JJ,K))/(SLAM(L,K)-SLAM(L-JJ,K)))      REVTAB
     *                  * (HV(L)-HV(L-JJ)) + HV(L-JJ)                   REVTAB
  500 CONTINUE                                                          REVTAB
  510 CONTINUE                                                          REVTAB
C                                                                       REVTAB
C  TEST FOR ASCENDING SWATH DATA NOT REQUIRED OR ALREADY GENERATED      REVTAB
C                                                                       REVTAB
      IF ( .NOT. LIDERG(4)) GO TO 575                                   REVTAB
      CALL PAGEP(1)                                                     REVTAB
      WRITE(KO,9000)                                                    REVTAB
      DO 550 MM = 1,181                                                 REVTAB
      CALL PAGEP(1)                                                     REVTAB
      WRITE(KO,9005) TIME(MM+1,KVEH),DLONG(MM+1,KVEH),                  REVTAB
     * TIME(MM+NADSCN,KVEH),DLONG(MM+NADSCN,KVEH),TIME(MM+NOSCN,KVEH),  REVTAB
     * DLONG(MM+NOSCN,KVEH),ALT(MM,KVEH),MM                             REVTAB
  550 CONTINUE                                                          REVTAB
  575 CONTINUE                                                          REVTAB
C                                                                       REVTAB
C                                                                       REVTAB
C                                                                       REVTAB
  600 CONTINUE                                                          REVTAB
C                                                                       REVTAB
C  SET MAXIMUM AND MINIMUM OF SWATH EDGES FOR TARGET PROCESSING         REVTAB
C                                                                       REVTAB
      MAXLT(KVEH) = 90 - MXLAT                                          REVTAB
      MINLT(KVEH) = 91 -MNLAT.                                          REVTAB
C                                                                       REVTAB
C                                                                       REVTAB
 9000 FORMAT(59H  REVTAB OUTPUT - TIME AND DELTA LONG FOR 3 POINTS, AND REVTAB
     *ALT )                                                             REVTAB
 9005 FORMAT( 7F14.7,I4)                                                REVTAB
 9010 FORMAT(21H 325/350 LOOP REVTAB ,6I6,2F15.7)                       REVTAB
 9020 FORMAT(I5,8F14.7)                                                 REVTAB
 9025 FORMAT(//)                                                        REVTAB
 9030 FORMAT(//27H MXLAT, MNMX, MXMN, MNLAT =,4I8)                      REVTAB
 9040 FORMAT(14H SLAM(361,L) =,8F14.8)                                  REVTAB
 9050 FORMAT(14H SLAM(362,L) =,8F14.8)                                  REVTAB
 9060 FORMAT(48H REVTAB ERROR - ANOMALY IN CROSS-TRACK LATITUDES)       REVTAB
 9070 FORMAT(1H ,2HK=,I2,1X,2HL=,I3,1X,3HAA=,F13.6,1X,3HJJ=,I1,1X,      REVTAB
     14HDEL=,F13.6,1X,3HXI=,E13.6,1X,4HINT=,I3,1X,5HNTRY=,I3,1X,        REVTAB
     24HLLL=,I3)                                                        REVTAB
C                                                                       REVTAB
      RETURN                                                            REVTAB
      END                                                               REVTAB
```

```
• FLT SWATH,1,751015, 4/511     • 1

      SUBROUTINE SWATH                                                  SWATH
C                                                                       SWATH
C                                                                       SWATH
C                                                                       SWATH
C     SUBROUTINE SWATH UTILIZES THE EPHEMERIS DATA FOR THE DAY SPECIFIESWATH
C     TO GENERATE THE LATITUDE AND DELTA LONGITUDE DATA FOR THE SCAN    SWATH
C     ANGLES FOR ONE REV. THE DATA IS GENERATED AT ONE DEGREE INCREMENTSSWATH
C     OF ECCENTRIC ANOMALY, BEGINNING AT THE ASCENDING NODE.            SWATH
      COMMON /HECCOM/                                                   HECCOM
     1ALPHA1,XBETAD(2),XBETA(2,2),XWD(2),XOMEGD(2),TASCNP(2),X10(2),    HECCOM
     2PERIOD(2),LNO,LDA,NREV1,NREV2,IYR,IYRB,IDAY,IPDAY,ICDAY,NREV1P,   HECCOM
     3NREV2P,IV1DAY,IV2DAY,LYR                                          SWATH
C
      COMMON /CONS/                                                     CONS
     1ECCF,CTS,CNUV,RADTDS,HEADER(20),NAME(2),NAME1(2),TWOPI,ALPHAM,    CONS
     2THES,WE,ORBIT1(6),ORBIT2(6),PI,CJ2,CMU,PI2,PI32,SA(3),RADTAN,     CONS
     3ICASE,IPPTS(10),NLINE,NPAGE,INP,LINMAX,ISWTH,ISWH,NVEH,NODAY,     CONS
     4IV1TIM(6),IV2TIM(6),KO,IORBIT,L10FBG(4)                           CONS
      REAL NAME,NAME1                                                   CONS
      LOGICAL L10FBG                                                    SWATH
C
      COMMON/ACCNUM/ACCNUM(100)                                         SWATH
      EQUIVALENCE (ACCNUM(  1),KVEH)                                    SWATH
     X,           (ACCNUM(  3),NUSCN)                                   SWATH
     X,           (ACCNUM(  4),NADSCN)                                  SWATH
     X,           (ACCNUM(  6),XN)                                      SWATH
     X,           (ACCNUM(  7),PER(1))                                  SWATH
     X,           (ACCNUM( 89),KDAY)                                    SWATH
     X,           (ACCNUM( 90),F)                                       SWATH
     X,           (ACCNUM( 91),TPER)                                    SWATH
C                                                                       SWATH
C                                                                       SWATH
      COMMON /ACCEPH/                                                   ACCEPH
     1INNODE(17,2),SH(1,2),EC(1,2),XIN(1,2),RTASC(1,2),WP(1,2),TNDE(1,2)ACCEPH
     2,RG(1)                                                            ACCEPH
      COMMON /ACCDAT/                                                   ACCDAT
     1TV(362),HV(362),SLAM(362,3),SPHT(362,3),ENP(2)                    ACCDAT
      COMMON /FLAG/                                                     FLAG
     1IFIRST,IVEH,ISTART,IEND,IDEND                                     FLAG
      DIMENSION ZB(3),YB(3),XNDE(2),P(3),R(3),PER(2)                    SWATH
      DIMENSION XL(3)                                                   SWATH
C                                                                       SWATH
C                                                                       SWATH
C  COMPUTE TRUE ANOMALY OF NORTHERNMOST POINT ON REV                    SWATH
      JDAY = 1                                                          SWATH
C                                                                       SWATH
      WNP = FMOD(TWOPI - WP(JDAY,KVEH) + PI2)                           SWATH
C                                                                       SWATH
C  SET CONSTANTS                                                        SWATH
C                                                                       SWATH
      ECC = EC(JDAY,KVEH)                                               SWATH
      TRM1 = 1. - ECC**2                                                SWATH
      TRM2 = SQRT(TRM1)                                                 SWATH
      TRM3 = TRM1/(1. + ECC*COS(WNP))                                   SWATH
C                                                                       SWATH
C  COMPUTE ECCENTRIC ANOMALY OF NORTHERNMOST POINT                      SWATH
C                                                                       SWATH
      SINE = TRM3*SIN(WNP)/TRM2                                         SWATH
```

```
      COSE = (1. - TRN3)/ECC                                            SWATH
      ENP(KVEH) = ACOS(COSE)                                            SWATH
      IF (SINE .LT. 0.0) ENP(KVEH) = TWOPI - ENP(KVEH)                  SWATH
C                                                                       SWATH
C                                                                       SWATH
C  COMPUTE RIGHT ASCENSION OF GREENWICH AND LONG. OF NODE AT NODE TIME  SWATH
C                                                                       SWATH
      AG = RG(JDAY) +(WE-XOMEGD(KVEH))*TNDE(JDAY,KVEH)                  SWATH
      XLONGN = RTASC(JDAY,KVEH) - AG                                    SWATH
      IF (XLONGN .LT. 0.0) XLONGN = XLONGN + TWOPI                      SWATH
C                                                                       SWATH
C                                                                       SWATH
C                                                                       SWATH
C                                                                       SWATH
C                                                                       SWATH
C                                                                       SWATH
C  COMPUTE POINTS AROUND REV                                            SWATH
C                                                                       SWATH
      IFL = 1                                                           SWATH
      DEL = 1./RADIAN                                                   SWATH
      EX  = E - DEL                                                     SWATH
      TPER = TPER - PER(KVEH)                                           SWATH
C                                                                       SWATH
      DO 400 I=1,362                                                    SWATH
C                                                                       SWATH
C                                                                       SWATH
      EX = EX + DEL                                                     SWATH
      IF(I .EQ. 362) GO TO 450
      IF(I .EQ. 362) GO TO 825
 40   TV(I) = (EX - ECC*SIN(EX))/XN + TPER                              SWATH
 45   IF (TV(I) .GE. TNDE(JDAY,KVEH)) GO TO 50                          SWATH
      TV(I) = TV(I) + PER(KVEH)                                         SWATH
      GO TO 45                                                          SWATH
C                                                                       SWATH
C  COMPUTE TRUE ANOMALY AT THIS TIME POINT ON REV                       SWATH
C                                                                       SWATH
 50   CONTINUE                                                          SWATH
      TV(I) = TV(I) - TNDE(JDAY,KVEH)                                   SWATH
      TRM4 = 1. - ECC*COS(EX)                                           SWATH
      SINV = TRM2*SIN(EX)/TRM4                                          SWATH
      COSV = (COS(EX) - ECC)/TRM4                                       SWATH
      V = ACOS(COSV)                                                    SWATH
      IF (SINV .LT. 0.0) V = TWOPI - V                                  SWATH
      TH = TV(I)                                                        SWATH
C                                                                       SWATH
C  COMPUTE INPLANE ANGLE FROM THE NODE AND VEHICLE ALTITUDE (E.R.)      SWATH
C                                                                       SWATH
      WT = V + WP(JDAY,KVEH)                                            SWATH
      IF (WT. GT. TWOPI) WT = WT - TWOPI                                SWATH
      HV(I) = SP(JDAY,KVEH)*TRM1/(1.+ECC*COS(V)) - 1.                   SWATH
      HT = HV(I)                                                        SWATH
C                                                                       SWATH
C  VEHICLE GEOCENTRIC LATITUDE AND FOR Z COMPONENT OF POSITION          SWATH
C                                                                       SWATH
      ZV = SIN(WT)*SIN(XIN(JDAY,KVEH))                                  SWATH
      VLAT = ASIN(ZV)                                                   SWATH
      SLAT(I,RADSCN) = VLAT*RADIAN                                      SWATH
C                                                                       SWATH
C  VEHICLE LONGITUDE AND FOR X,Y COMPONENTS OF POSITION                 SWATH
C                                                                       SWATH
```

```
      AA = 0.                                                           SWATH
      IF (PI2 .LE. WT .AND. WT .LE. PI32) AA = 1.                       SWATH
      TEMP = XLONGN -(WE-XOMEGD(KVEH))*TV(I)                            SWATH
      IF ( T .LT. 361) GO TO 65                                         SWATH
      AA = 0.                                                           SWATH
      TEM = SIGN(PI2,PI2-XIN(JDAY,KVEH))                                SWATH
      IF ( T .EQ. 361) TEM = -TEM                                       SWATH
      GO TO 70                                                          SWATH
 65   TEM = ATAN(TAN(WT)*COS(XIN(JDAY,KVEH)))                           SWATH
 70   XLON = TEMP + TEM - AA*PI                                         SWATH
      XV = COS(VLAT)*COS(XLON)                                          SWATH
      YV = COS(VLAT)*SIN(XLON)                                          SWATH
      DXLON = XLON - XLONGN                                             SWATH
      SPHI(I,NADSCN) = DXLON                                            SWATH
      IF (SPHI(I,NADSCN) .GT. PI2) SPHI(I,NADSCN) = SPHI(I,NADSCN)      SWATH
     X  - TWOPI                                                         SWATH
C                                                                       SWATH
C  Z BODY AXIS IS NEGATIVE OF VEHICLE POSITION VECTOR                   SWATH
C                                                                       SWATH
      ZB(1) = -XV                                                       SWATH
      ZB(2) = -YV                                                       SWATH
      ZB(3) = -ZV                                                       SWATH
C                                                                       SWATH
C  INSTANTANEOUS NODE VECTOR COMPONENTS (ECR)                           SWATH
C                                                                       SWATH
      XNDE(1) = COS(TEMP)                                               SWATH
      XNDE(2) = SIN(TEMP)                                               SWATH
C                                                                       SWATH
C  Y BODY AXIS NODE VECTOR CROSSED INTO Z-BODY AXIS                     SWATH
C                                                                       SWATH
      YB(1) = XNDE(2)*ZB(3)                                             SWATH
      YB(2) = -XNDE(1)*ZB(3)                                            SWATH
      YB(3) = XNDE(1)*ZB(2) - XNDE(2)*ZB(1)                             SWATH
C                                                                       SWATH
C  MAKE Y BODY AXIS A UNIT VECTOR AND CHANGE SIGN IF VEHICLE IS IN      SWATH
C  SOUTHERN HEMISPHERE                                                  SWATH
C                                                                       SWATH
      TEMP = SQRT(YB(1)**2 + YB(2)**2 + YB(3)**2)                       SWATH
      IF( VLAT .LT. 0.0) TEMP = -TEMP                                   SWATH
      DO 75   J = 1,3                                                   SWATH
 75   YB(J) = YB(J)/TEMP                                                SWATH
C                                                                       SWATH
C  COMPUTE VEHICLE POSITION VECTOR IN ECR COMPONENTS                    SWATH
C                                                                       SWATH
      D = 1. + HV(I)                                                    SWATH
      DO  100 J = 1,3                                                   SWATH
 100  P(J) = -D*ZB(J)                                                   SWATH
C                                                                       SWATH
C                                                                       SWATH
C                                                                       SWATH
C                                                                       SWATH
C                                                                       SWATH
C                                                                       SWATH
C                                                                       SWATH
C  COMPUTE SWATH DATA FOR EACH SCAN ANGLE                               SWATH
C                                                                       SWATH
      DO 300  K = 1,NOSCN                                               SWATH
C                                                                       SWATH
      IF (SA(K) .EQ. 0.0) GO TO 300                                     SWATH
      XLY = SIN(SA(K))                                                  SWATH
      XLZ = COS(SA(K))                                                  SWATH
```

```
C                                                                       SWATH
C  GENERATE COMPONENTS OF ECR DIRECTION VECTOR AT CURRENT SCAN ANGLE    SWATH
C                                                                       SWATH
      DO 150  L = 1,3                                                   SWATH
 150  XL(L) = XLY*YB(L) + XLZ*ZB(L)                                     SWATH
C                                                                       SWATH
C  GENERATE COMPONENT B (I.E., DOT PRODUCT OF VECTORS XL AND P)         SWATH
C                                                                       SWATH
      XLMAG = SQRT(XL(1)**2 + XL(2)**2 + XL(3)**2)                      SWATH
      B = 0.0                                                           SWATH
      DO 200  L = 1,3                                                   SWATH
 200  B = B + XL(L)*P(L)                                                SWATH
      B = B/XLMAG                                                       SWATH
C                                                                       SWATH
C  COMPUTE DISTANCE S                                                   SWATH
C                                                                       SWATH
      IF ((B*B - D*D + 1.) .GE. 0.0) GO TO 225                          SWATH
C                                                                       SWATH
      GO TO 295                                                         SWATH
C                                                                       SWATH
C                                                                       SWATH
 225  CONTINUE                                                          SWATH
C                                                                       SWATH
      S = B + SQRT(B*B - D*D + 1.)                                      SWATH
C                                                                       SWATH
C  TEST FOR INTERSECTION OF THIS SCAN VECTOR WITH EARTH. IF NONE GO 300 SWATH
C                                                                       SWATH
      IF ( S .GT. 0.0) GO TO 295                                        SWATH
C                                                                       SWATH
C  COMPUTE ECR COMPONENTS OF VECTOR FROM EARTH CENTER TO INTERSECTION PTSWATH
C                                                                       SWATH
      DO 250  L = 1,3                                                   SWATH
 250  R(L) = P(L) - S*XL(L)/XLMAG                                       SWATH
C                                                                       SWATH
C  COMPUTE GEOCENTRIC LAT. AND DELTA LONG. FOR THIS INTERSECTION POINT  SWATH
C                                                                       SWATH
      IF (ABS(R(3)) .LE. 1.0) GO TO 275                                 SWATH
      RTEMP = SIGN(1.,R(3))                                             SWATH
      R(3) = RTEMP                                                      SWATH
C                                                                       SWATH
C                                                                       SWATH
 275  CONTINUE                                                          SWATH
      SLAM(I,K) = ASIN(R(3))*RADIAN                                     SWATH
      TEMP = ATAN2(R(2),R(1))                                           SWATH
      TEM = PIMOD(TEMP) - ALONGN                                        SWATH
      IFLG = 0                                                          SWATH
 280  IF ( ABS(TEM-DXLON) .LT. PI2) GO TO 290                           SWATH
      IF ( IFLG .GT. 5) GO TO 288                                       SWATH
      IF (TEM .LT. 0.) GO TO 285                                        SWATH
      IFLG = IFLG + 1                                                   SWATH
      TEM = TEM - TWOPI                                                 SWATH
      GO TO 280                                                         SWATH
 285  TEM = TWOPI - TEM                                                 SWATH
      IFLG = IFLG + 1                                                   SWATH
      GO TO 280                                                         SWATH
 288  CALL EXERR(1)                                                     SWATH
      WRITE(KD,9010) XLONGN,TEM,DXLON,R(1),R(2),IFLG                    SWATH
 290  CONTINUE                                                          SWATH
      SPHI(I,K) = TEM                                                   SWATH
      GO TO 300                                                         SWATH
```

```
C                                                                       SWATH
C  ZERO LATITUDE AND LONGITUDE DIFFERENCE VALUES FOR THIS SCAN ANGLE    SWATH
C                                                                       SWATH
 295  CONTINUE                                                         SWATH
      SLAM(I,K) = 1.0E6                                                 SWATH
      SPHI(I,K) = 1.0E6                                                 SWATH
C                                                                       SWATH
 300  CONTINUE                                                         SWATH
C                                                                       SWATH  **-1
C  COMPUTE DATA AT N. P. AND S. P.                                      SWATH
C                                                                       SWATH
      GO TO 400                                                         *NEW
 425  CONTINUE                                                         SWATH  **-1
      EX = FNP(KVEH)                                                    SWATH
      IFL = 2                                                           SWATH
      GO TO 40                                                          SWATH
 450  CONTINUE                                                         SWATH
      WSP = WNP + PI                                                    SWATH
      TRM3 = TRM1/(1. + ECC*COS(WSP))                                   SWATH
      SINF = TRM3*SIN(WSP)/TRM2                                         SWATH
      COSF = (1. - TRM3)/ECC                                            SWATH
      EX = ACOS(COSF)                                                   SWATH
      IF ( SINE .LT. 0.0) EX = TWOPI - EX                               SWATH
      IFL = 3                                                           SWATH
      GO TO 40                                                          SWATH
 400  CONTINUE                                                          *NEW
C                                                                       SWATH
C                                                                       SWATH
C                                                                       SWATH
 600  CONTINUE                                                         SWATH
      IF ( .NOT. L1DEBG(1)) GO TO 620                                   SWATH
C                                                                       SWATH
      NADSC = NADSCN-1                                                  SWATH
      CALL EJECT(2)                                                     SWATH
      WRITE( 6,9000)                                                    SWATH
      DO 610 I=1,362                                                    SWATH
      CALL PAGER(1)                                                     SWATH
      WRITE(KD,9005)((SLAM(I,J),SPHI(I,J),J=1,NOSCN),HV(I),TV(I),I)     SF12
  610 CONTINUE                                                          SWATH
C                                                                       SWATH
C                                                                       SWATH
 9000 FORMAT(62H SWATH OUTPUT- LAT. AND DELTA LONG FOR 3 POINTS, ALT ANDSWATH
     X TIME )                                                           SWATH
 9005 FORMAT( 8F14.7,I4)                                                SWATH
 9010 FORMAT(26H ITERATION LIMIT ON SWATH ,5F15.8,I3)                   SWATH
  620 RETURN                                                            SWATH
      END                                                               SWATH
```

```
* FLT WRITSW,1,750925, 58804    , 1

000001          SUBROUTINE WRITSW(IOPT)                                         WRITSW
000002    C     THIS ROUTINE INITIALIZES,WRITES AND TERMINATES OUTPUT RECORDS ON WRITSW
000003    C     THE SWATH TABLE AN SWATH REFERENCE FILES.                       WRITSW
000004    C     NOTE THIS ROUTINE MUST BE RECODED FOR UNIVAC AND THE PDP.       WRITSW
000005          COMMON /CONS/                                                   CONS
000006         1ECCE,CIS,CNUV,RADIUS,HEADER(20),NAME(2),NAME1(2),TWOPI,ALPHAM,  CONS
000007         2THES,WE,ORBIT1(6),ORBIT2(6),PI,CJ2,CNU,PI2,PI32,SA(5),RADIAN,   CONS
000008         3ICASE,IPPTS(10),NLINL,NPAGE,INP,LINMAX,ISWIN,ISWR,NVEH,NUDAY,   CONS
000009         4IV1TIM(6),IV2TIM(6),KO,TORBIT,LIDEBG(4)                         CONS
000010          REAL NAME,NAME1                                                 CONS
000011          LOGICAL LIDEBG                                                  CONS
000012          COMMON /HECCOM/                                                 HECCOM
000013         1ALPHAT,XBETAD(2),XBETA(2,2),XHD(2),XONEGD(2),TASCNP(2),XTO(2),  HECCOM
000014         2PERTOD(2),LMU,LDA,NREV1,NREV2,IYR,IYRB,IDAY,IPDAY,ICDAY,NREV1P, HECCOM
000015         3NREV2P,IV1DAY,IV2DAY,LYP                                        SET1
000016          COMMON /ACCEPP/                                                 ACCEPH
000017         1TMNODE(12,2),SM(1,2),EC(1,2),XIN(1,2),RTASC(1,2),WP(1,2),TNDE(1,2)ACCEPH
000018         2,RG(1)                                                          ACCEPH
000019          COMMON /NREV/                                                   NREV
000020         1XLONG(1,2),NREV(2)                                              NREV
000021          DIMENSION NREVCT(1,2)                                           NREV
000022          EQUIVALENCE (NREV(1),NREVCT(1,1))                               NREV
000023          COMMON /SWAT/                                                   SWAT
000024         1DLONMN(100),DLONMX(100),TIME1(100,3),DLONG1(100,3),INLAT(2),    SWAT
000025         2ISLAT(2),ISOSTR,NLAT,LATND(100),IALT(100),ICRDI(17)             SWAT
000026          COMMON/ACCNUM/ ACCNUM(100)                                      WRITSW
000027          EQUIVALENCE (ACCNUM(1),KVEH),(ACCNUM(89),KUDAY)                 WRITSW
000028          COMMON/TLARUS/ CARUS(8,17,2)                                    WRITSW
000029          COMMON /FLAG/                                                   FLAG
000030         1IFIRST,IVEH,ISTART,IEND,IDEND                                   FLAG
000031          COMMON /XDAY/ IDAS(2)                                           SET7
000032          DIMENSION A(800),XNLAT(2),XSLAT(2),V1TIM(3),V2TIM(3)            WRITSW
000033          DIMENSION ALTI(100),ATND(100)                                   WRITSW
000034          EQUIVALENCE (ALTI(1),IALT(1))(ATND(1),LATND(1))                 WRITSW
000035          EQUIVALENCE(XCASE,ICASE),(XVEH,NVEH),(XNUDAY,NUDAY),( XLAT,NLAT), WRITSW
000036         1(XNLAT(1),INLAT(1))(XSLAT(1),ISLAT(1)),(V1TIM(1),IV1TIM(1)),     WRITSW
000037         2(V2TIM(1),IV2TIM(1)),(SOSTR,ISOSTR)                             WRITSW
000038          DO 5 I=1,800                                                    WRITSW
000039          A(I) = 0.                                                       WRITSW
000040        5 CONTINUE                                                        WRITSW
000041          IF(IOPT .EQ. 2)GO TO 100                                        WRITSW
000042          IF(IOPT .EQ. 3)GO TO 200                                        WRITSW
000043          IF(IOPT .EQ. 4)GO TO 300                                        WRITSW
000044          DEFINE FILE 8(1101,800,U,IDUM)
000045          A(1) = NAME(1)                                                  WRITSW
000046          A(2) = NAME(2)                                                  WRITSW
000047          A(3) = XCASE                                                    WRITSW
000048          A(4) = XVEH                                                     WRITSW
000049          A(5) = XNUDAY                                                   WRITSW
000050          A(7) = XNLAT(1)                                                 WRITSW
000051          A(8) = XNLAT(2)                                                 WRITSW
000052          A(9) = XSLAT(1)                                                 WRITSW
000053          A(10)= XSLAT(2)                                                 WRITSW
000054          DO 8 I=1,3                                                      WRITSW
000055          K = I + 10                                                      WRITSW
000056          L = I +13                                                       WRITSW
000057          A(K) =  V1TIM(I)                                                WRITSW
000058          A(L) =  V2TIM(I)                                                WRITSW
```

ORIGINAL PAGE IS OF POOR QUALITY

```
000059      8 CONTINUE                                                    WRITSW
000060        A(17) = SOSTR                                               WRITSW
000061        A(6) = XLAT                                                 WRITSW
000062        NOWDS = 8*NLAT                                              WRITSW
000063        FIND(ISWTB'1)
000064        WRITE(ISWTB'1,ERR=600) (A(I),I=1,NOWDS)
000065        REWIND ISWR                                                 WRITSW
000066        WRITE(ISWR) NAME1(1),NAME1(2),ICASE,NVEH,NODAY,(IV1TIM(I),I=1,3), WRITSW
000067       1(IV2TIM(I),I=1,3),((CARUS(I,1,J),J=1,2),I=1,3),XOMEGD(1),XOMEGD(2)WRITSW
000068       2,XWD(1),XWD(2),(SA(I),I=1,3),(A(I),I=23,54)                  SET3
000069        GO TO 500                                                   WRITSW
000070      100 IS1 = NLAT                                                WRITSW
000071        IS2 = 2*NLAT                                                WRITSW
000072        IS3 = 3*NLAT                                                WRITSW
000073        DO 110 I=1,NLAT                                             WRITSW
000074        A(I) =  ATWD(I)                                             WRITSW
000075        A(IS1+I) = ALTI(I)                                          WRITSW
000076      110 CONTINUE                                                  WRITSW
000077        DO 130 J=1,3                                                WRITSW
000078        IPT = (J-1)*NLAT                                            WRITSW
000079        DO 120 I=1,NLAT                                             WRITSW
000080        A(IS2+IPT+I) = TIME1(I,J)                                   WRITSW
000081        A(IS3+IPT+I) = DLONG1(I,J)                                  WRITSW
000082      120 CONTINUE                                                  WRITSW
000083      130 CONTINUE                                                  WRITSW
000084        IWRS = TEND - IDAS(1) + 1                                   SET7
000085        IREC = (IWRS +1)*(KVEH-1) + KDAY + 1                        SET7
000086        FIND(ISWTB'IREC)
000087        WRITE(ISWTB'IREC,ERR=600) (A(I),I=1,NOWDS)
000088        WRITE(ISWR)KVEH,KDAY,WP(1,KVEH),NREV(KVEH),(IMNODE(I,KVEH),I=1,17)SET1
000089       1,                                                           SET1
000090       1(CARUS(4,I,KVEH),I=1,17),(IGNDI(I),I=1,17)                  WRITSW
000091        IF(.NOT.LTDEBG(2))GO TO 150                                 SET1
000092        NR = NREV(KVEH)                                             SET2
000093        CALL PAGER(NR+4)                                            SET2
000094        WRITE(KO,900)KVEH,KDAY                                      SET1
000095      900 FORMAT(1H0,27X,32HSWATH REFERENCE RECORD   VEHICLE,1X,I1,3X,3HDAY,SET1
000096       11X,I3)                                                      SET2
000097        WRITE(KO,901)                                               SET1
000098      901 FORMAT(1H0,33X,38HWP        NREV       IMNODE        CARUS)SET2
000099        WRITE(KO,902) WP(1,KVEH),((I,IMNODE(I,KVEH),CARUS(4,I,KVEH)),I=1, SET1
000100       1NR)                                                         SET1
000101      902 FORMAT(1H ,27X,F13.6,3X,I2,4X,E12.6,2X,F13.6//,(1H+,43X,I2,4X, SET3
000102       1E12.6,2X,F13.6/))                                           SET3
000103        CALL PAGER(NLAT+4)                                          SET2
000104        WRITE(KO,903)KVEH,KDAY                                      SET1
000105      903 FORMAT(1H0,27X,28HSWATH TABLE RECORD   VEHICLE,1X,I1,3X,3HDAY,1X, SET1
000106       1I3)                                                         SET2
000107        WRITE(KO,904)                                               SET1
000108      904 FORMAT(1H0,13X,80H LATNO  IALT  TIME(1)  TIME(2)  TIME(3)    DLONGSET2
000109       1(1)     DLONG1(2)     DLONG1(3))                            SET2
000110        WRITE(KO,905)((LATNO(I),IALT(I),(TIME1(I,J),J=1,3),(DLONG1(I,J), SET1
000111       1J=1,3)),I=1,NLAT)                                           SET1
000112      905 FORMAT(1H ,15X,I3,2X,I5,3X,F5.0,4X,F5.0,4X,F5.0,3X,                *NEW
000113       1  E12.6,2X,E12.6,2X,E12.6)                                        *NEW
000114      150 CONTINUE                                                  SET1      **-1
000115        GO TO 500                                                   WRITSW
000116      200 DO 210 I=1,NLAT                                           WRITSW
000117        A(I) = ATWD(I)                                              SET7
000118        A(I+NLAT) = DLONNN(I)                                       SET3
```

```
000119          A(I+2*NLAT) = DLONMX(I)                                  SET3
000120      210 CONTINUE                                                 WRITSW
000121          IWR1 = IEND - IDAS(1) + 1                                SET7
000122          IWR3 = IEND - IDAS(KVEH) + 1                             SET7
000123          IRECN = (IWR1+1)*(KVEH-1) + IWR3 + 2                     SET7
000124          FIND(ISWTB'IRECN)
000125          WRITE(ISWTB'IRECN,ERR=600) (A(I),I=1,NOWDS)
000126          IF(.NOT. DIDEBG(2))GO TO 500                             SET2
000127          CALL PAGEP(NLAT+4)                                       SET2
000128          WRITE(KO,910)KVEH                                        SET2
000129      910 FORMAT(1H0,27X,34HSWATH TABLE SEARCH RECORD VEHICLE ,I3,I1)  SET2
000130          WRITE(KO,915)                                            SET2
000131      915 FORMAT(1H0,27X,37HLATNO      DLONMX                 DLONMN)  SET2
000132      920 FORMAT(1H ,20X,I5,3X,F13.6,9X,F13.6)                     SET2
000133          WRITE(KO,920)(LATNO(I),DLONMX(I),DLONMN(I),I=1,NLAT)     SET2
000134          GO TO 500                                                WRITSW
000135      300 CONTINUE                                                 WRITSW
000136          END FILE ISWB                                            WRITSW
000137          REWIND ISWB                                              WRITSW
000138      500 RETURN
000139      600 WRITE(KO,930)
000140      930 FORMAT(1H0,83H *** AN IRRECOVERABLE I/O ERROR HAS OCCURRED ON WRIT
000141         1ING A RECORD ON THE SWATH TABLE   /10X,31H THE JOB IS BEING ABANDO
000142         2NED ***)
000143          STOP                                                     WRITSW
000144          END
```

3. TOC

GRID BOOK II

Table of Contents

PART I

PROBLEM DESCRIPTION

UTILITY PROGRAM GRID

1.0 SCOPE

1.1 PROGRAM CAPABILITIES

This utility processes NASA's climatology tape to obtain the weather index along with its associated latitude and longitude. It then converts the latitude and longitude to a grid no pair (I, J). Using J as a record number, the utility writes a record onto an index matrix location file. Each record contains all the weather indexes associated with a given J value.

1.2 OPERATIONAL ASSUMPTIONS

- The weather index is in the range from 1 to 16000.

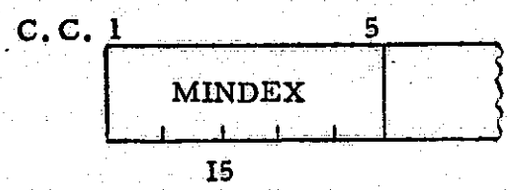
2.0 INPUT

2.1 CARDS

2.1.1 Data Quantities

MINDEX - maximum number of index points to process on the NASA climatology tape. Range 1 to 16000, nominal value is 16000 (used to reduce run time for debugging cases).

2.1.2 Card Format

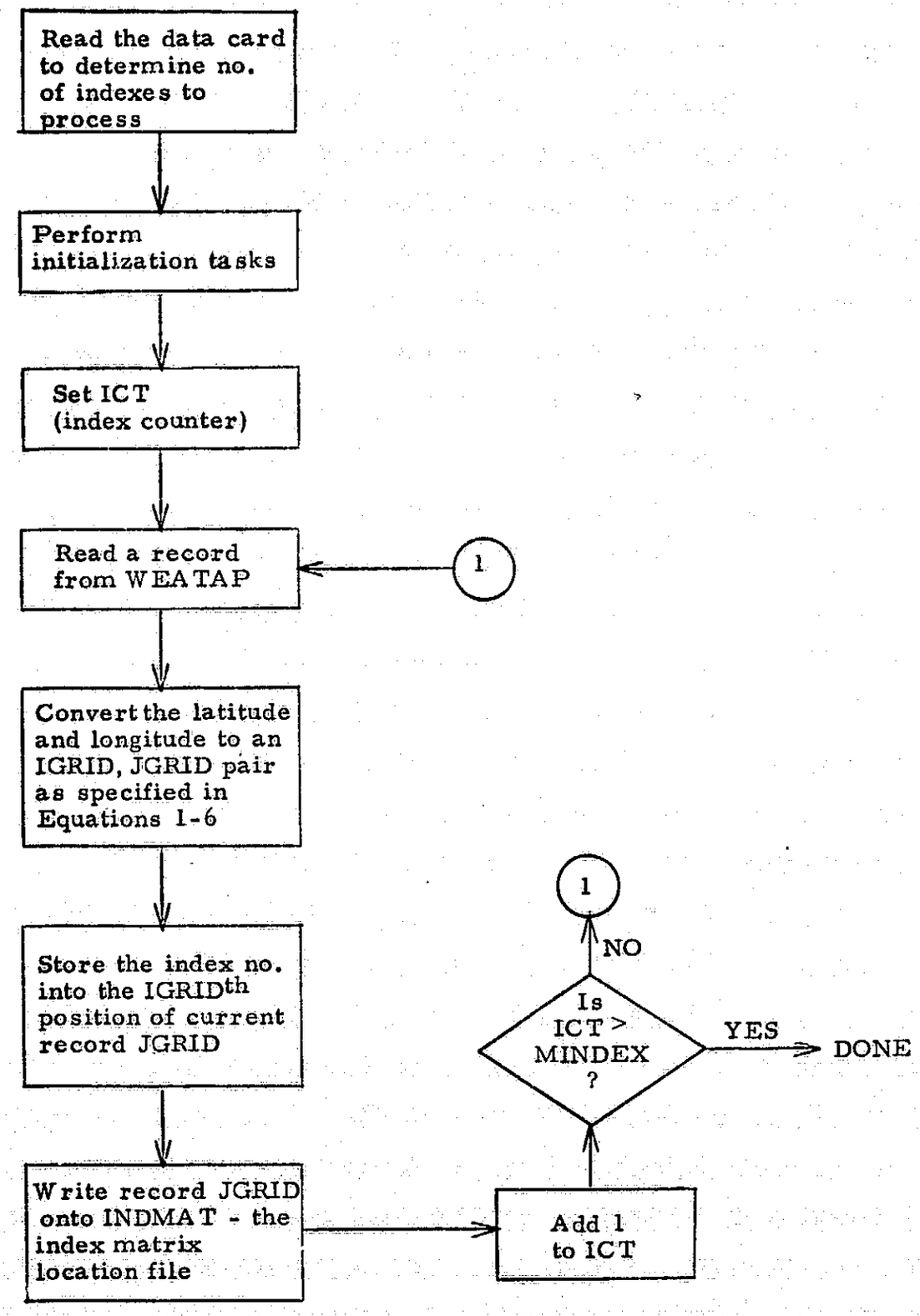


2.2 FILES

The only input file is the NASA climatology tape assigned to WEATAP - unit 13. See Section 2.4 of the Users Manual for a description of the file.

3.0 PROCESSING

3.1 PROGRAM FLOW



3.2 EQUATIONS

Given ϕ - latitude in radians

λ - longitude in radians

$I\phi = 125$, $J\phi = 125$ - North Pole index set

Compute stereographic projection

1. $\phi' = ABS(\phi)$
2. $R = 124.817494012 * \tan\left(\frac{\pi - 2 * \phi'}{4}\right)$
3. $A = \lambda - \pi/18$
4. $FI = I\phi + R * \cos(A)$
   $FJ = J\phi - R * \sin(A)$
5. If ϕ is minus $FI = FI + 2 * I\phi$
6. $IGRID = FI + .5$
   $JGRID = FJ + .5$

IGRID - the horizontal axis grid coordinate

JGRID - the vertical axis grid coordinate

4.0 OUTPUT

4.1 PRINT DATA

There is no normal printed output.

4.2 FILES

The only output file is INDMAT - the index matrix location file. See Section 2.4 of the Users Manual for a description of its contents.

5.0 ERROR PROCESSING

5.1 GENERAL

There are only three checks made for errors, one of which is fatal.

5.2 ERROR CHECKS AND MESSAGES

Fatal

1. A check is made to make sure MINDEX on the input card is between 0 and 16000. If not, the following message prints and the program stops.

   Message:

   MINDEX IS NOT BETWEEN 0 AND 16000. PROGRAM IS TERMINATED.

Non-Fatal

2. A check is made to make sure the index no. read from WEATAP file is between 1 and 16000. If not, the index no. is not included in the current INDMAT file record and the program reads the next WEATAP record.

   Message:

   THE INDEX NO. READ FROM WEATAP IS NOT BETWEEN 1 AND 16000. THE DATA IS IGNORED.

3. A check is made to see if an IGRID, JGRID grid pair has already been assigned an index number. If so, a message is printed and the current index no. replaces the one originally stored.

   Message:

   THE GRID PAIR IGRID nnn JGRID nnn HAS AN INDEX NUMBER ALREADY ASSIGNED.

PART II

COMMON BLOCK DEFINITIONS

COMMON STORAGE ALLOCATION

Name DIRAC Size _____ Page _____ of _____

Function Contains data for INDMAT direct access file

Name	Dimension	Format	Description	Symbol	Units
ICDCIN	252	I	Index array for CDC to support INDMAT file		
IREC	500	I	One record for INDMAT file current record to be read/written		
IGRID	1	I	The position in IREC in which to store IGRPT (row no.)		
JGRID	1	I	The current column ≡ rec. no. to read or write		
JSAVE	1	I	The previous value of JPT		
MAXJ	1		= 41 for hand off case test = 251 for final version		

COMMON STORAGE ALLOCATION

Name WEATDT Size _____ Page _____ of _____

Function Contains data needed from weather tape

Name	Dimen-sion	For-mat	Description	Sym-bol	Units
IGRDPT	1	I	Index no. - 1 to 16000		
LAT	4	I, A	LAT(1) + deg, LAT(2) - min LAT(3) - sec, LAT(4) - S or N		
LON	4	I, A	LON(1) - deg, LON(2) - min LON(3) - sec, LON(4) - E or W		
IA	(500)	I	Weather tape input record area - physical		
IB	(50)	I	One logical UNIVAC tape record		
IC	(30)	I	One logical CDC tape record		

COMMON STORAGE ALLOCATION

Name FLAG Size _____ Page _____ of _____

Function _____

Name	Dimen-sion	For-mat	Description	Sym-bol	Units
INP	1	I	Input card unit ≡ S		
IOUT	1	I	Print unit ≡ 6		
IWTAPE	1	I	Weather tape unit ≡ 13		
IGRDTP	1	I	Grid matrix file unit ≡ 4		

PART III

LIST OF SUBROUTINES

AND

SUBROUTINE CALL STRUCTURE

LIST OF SUBROUTINES IN GRID

	Name	Function
1.	GRID	Main control routine for GRID program. It reads the input control card, controls the reading of the weather tape, computes latitude and longitude and controls the writing of the index matrix file.
2.	RDGRID	Reads one record from the index matrix file.
3.	WRGRID	Writes the index matrix file.
4.	RDWETR	Reads the NASA weather tape.

SUBROUTINE CALL STRUCTURE

GRID

- WRGRID
- RDWETR
- RDGRID

PART IV

SUBROUTINE DESCRIPTION

AND

FLOWCHARTS

SUBROUTINE GRID

Purpose:

This is the main control routine for the GRID program. It reads the control card, controls the reading of the weather tape, computes from the latitude and longitude associated with a weather grid the I, J coordinates of the index matrix and controls the writing of the index matrix file.

Input:

The quantity MINDEX is obtained from the control card.

IGROPT, LAT and LON are obtained from the file IWTAPE. INP and IOUT from FLAG COMMON.

Output:

MAXJ records on file IGRDTP.

Linkage:

None.

Subroutines Used:

CALL RDWETR(IRC) - IRC is relative logical Rec. No. within a physical record. Initialized to 1 first time only.

CALL RDGRID - Read 1 record JGRID from IGRDTP file.

CALL WRGRID(IOP) - IOP = 0. Open IGRDTP file and initialize all MAXJ rec. to 0.

IOP = +N. Write record from IREC to IGRDTP file.

IOP = -N. Write record and close the IGRDTP file.

Local Variables:

INDEX - Count on no. of weather index rec. to read from IWTAPE

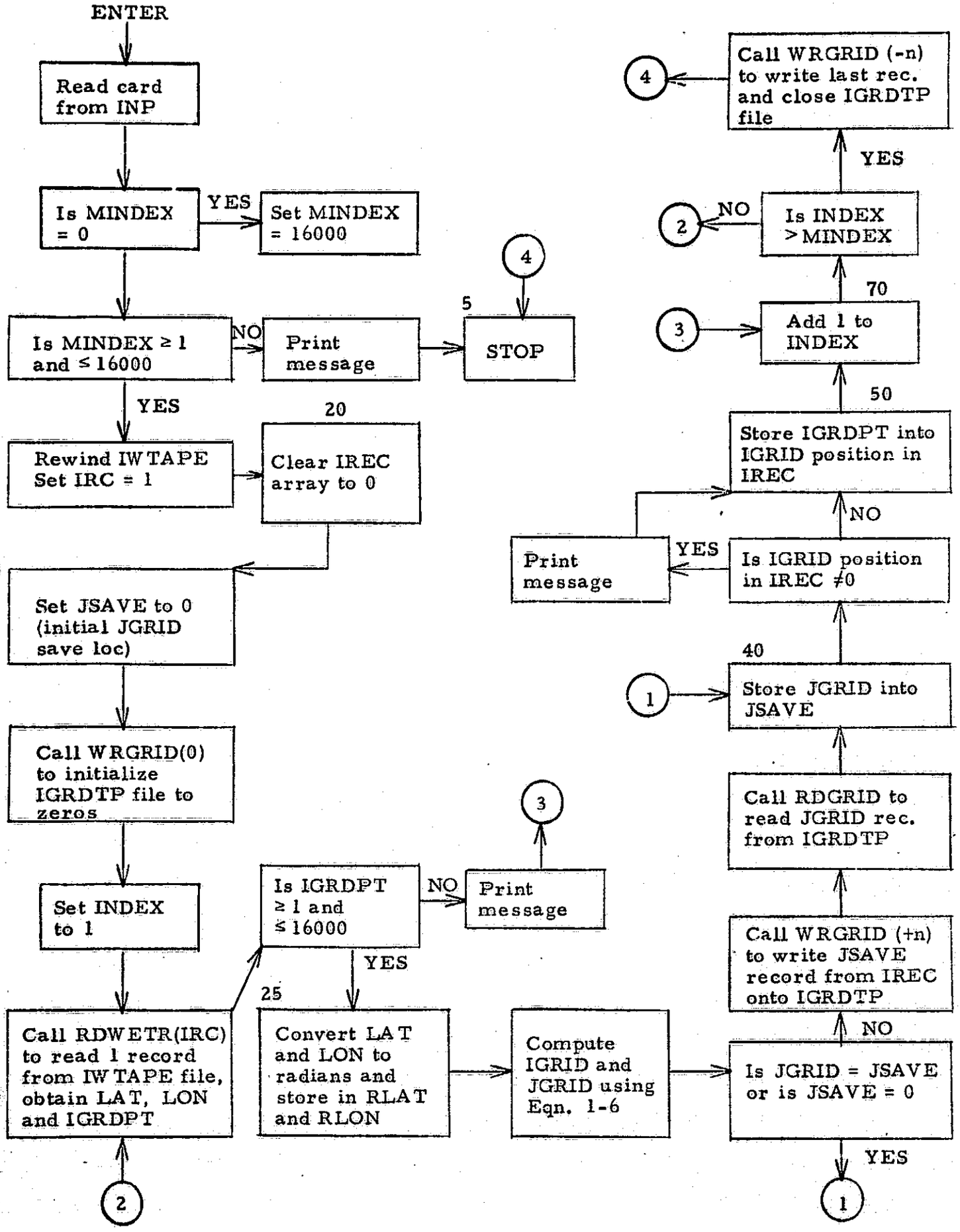
RLAT - Latitude in radians

RLON - Longitude in radians

Processing:

See flowchart for logic flow.
Equations 1-6 are described in the Problem Description.

Subroutine GRID Block Diagram



SUBROUTINE RDGRID

Purpose:

To read one record from the direct access file IGRDTP. The Record No. to be read is stored in JGRID and the record is read into IREC. For the CDC, READMS is to be used and for the UNIVAC, the direct access read. The record length is 500 words.

Input:

ICDCIN, JGRID in DIRAC COMMON.
IGRDTP in FLAG COMMON.
The data record on IGRDTP file.

Linkage:

CALL RDGRID.

Subroutines Used:

None.

Processing:

Give a call to READMS (CPC) or read from unit IGRDTP for UNIVAC.

SUBROUTINE WRGRID

Purpose:

On option, create a NULL IGRDTP D.A. file, or write one record at the JSAVE location from IREC or write a record and close the file. The record length is 500 words.

Input:

ICDCIN, IREC, JSAVE, MAXJ, JGRID in DIRAC COMMON.
IGRDTP in FLAG COMMON.

Output:

Either a NULL file on IGRDTP or one record in the JSAVE loc. of IGRDTP from IREC.

Linkage:

CALL WRGRID(IOP)

IOP = 0 - Generate NULL file.
IOP = +n - Write one record.
IOP = -n - Write one record and close the file.

Subroutines Used:

None.

Processing:

IOP = 0 - Give call to OPENMS for CDC and define file for UNIVAC (MAXJ gives no. of records), and then write MAXJ records from IREC using WRITEMS or WRITE (IREC has been cleared to zero).

IOP = +n - Give call to WRITEMS or write to WRITE IREC on JSAVE record.

IOP = -n - Give call to WRITEMS or write to WRITE IREC on JGRID record. Give call to CLOSEMS or call CLOSE (IGRDTP, 0) to close the file.

SUBROUTINE RDWETR

Purpose:

This routine reads one physical record from the NASA weather tape. It unblocks the record and stores one logical record into working storage.

Input:

One physical record on IWTAPE unit.

COMMON FLAG
IWTAPE

COMMON WEATDT
IA, IB

Output:

COMMON WEATDT
IGRDPT, LAT, LON

Linkage:

CALL RDGRID(IRC). IRC is a logical record count 1 to 10 initialized to 1 by the calling routine for first call only. Routine then adds 1 to it and resets it to 1 as needed.

Method/Procedure:

Use NTRAN to perform the read from the weather tape.

Subroutines Called:

NTRAN.

PART V

SUBROUTINE LISTINGS

```
* ELT BLKDAT,1,760401, 68088    , 1

000001          BLOCK DATA                                                  BLOCK
000002          COMMON /FLAG/                                               FLAG
000003         1INP,IOUT,IWTAPE,IGRDTP                                      FLAG
000004          COMMON /DTRAC/                                              DTRAC
000005         1ICDCIN(252),IREC(500),IGRID,JGRID,JSAVE,MAXJ                DEBUG1
000006          COMMON /WEATDT/                                             WEATDT
000007         1IGRDPT,LAT(4),LON(4),IA(500),IB(50 ),IC(50)
000008          DATA MAXJ/42/,INP/5/,IOUT/6/,IWTAPE/13/,IGRDTP/4/           DEBUG1
000009          END                                                         BLOCK
```

```
@ELT GRID,1,760401, 68090      , 1

      C     PROGRAM GRID(ITEST=401,OUTPUT=401,TAPE5=ITEST,TAPE6=OUTPUT,        DEBUG1
      C    1INDMAT=1004,WEATAP=1004,TAPE4=INDMAT,TAPE13=WEATAP)                GRID
           COMMON /FLAG/                                                       FLAG
           1IMP,IOUT,IWTAPE,IGRDTP                                             FLAG
           COMMON /WEATDT/                                                     WEATDT
           1IGRDPT,LAT(4),LON(4),IA(500),IB(50 ),IC(30)
           COMMON /DTRAC/                                                      DTRAC
           1ICDCIN(252),IREC(500),IGRID,JGRID,JSAVE,MAXJ                       DEBUG1
           DIMENSION SLAT(3),SLON(3)                                           GRID
           DATA DEGTOR/.01745329/,I0/125/,J0/125/,PI/3.1415926/                GRID
           READ(IMP,900)MINDEX                                                 GRID
           IF(MINDEX .EQ. 0)MINDEX = 16000                                     GRID
           IF(MINDEX .GE. 1 .AND.MINDEX .LE. 16000)GO TO 10                    GRID
           WRITE(IOUT,901)                                                     GRID
         5 STOP                                                                GRID
        10 CALL NTRAN (IWTAPE, 10, 22)                                         GRID
           IRC = 1                                                             GRID
           DO 20 I=1,500                                                       GRID
           IREC(I) = 0                                                         GRID
        20 CONTINUE                                                            GRID
           JSAVE = 0                                                           GRID
           CALL WRGRID(0)                                                      GRID
           DO 70 I=1,MINDEX                                                    GRID
           CALL RDWETR(IRC)                                                    GRID
           IF(IGRDPT .GE. 1.AND. IGRDPT .LE. 16000)GO TO 25                    GRID
           WRITE(IOUT,902)                                                     GRID
           GO TO 70                                                            GRID
        25 DO 30 J=1,3                                                         GRID
           SLAT(J) = LAT(J)                                                    GRID
           SLON(J) = LON(J)                                                    GRID
        30 CONTINUE                                                            GRID
           RLAT = SLAT(1) + SLAT(2)/60. + SLAT(3)/3600.                        GRID
           RLON = SLON(1) + SLON(2)/60. + SLON(3)/3600.                        GRID
           IF(LAT(4) .EQ. IHS)RLAT = -RLAT                                     GRID
           IF(LON(4) .EQ. IHW)RLON = 360. -RLON                                GRID
           RLAT = DEGTOR*RLAT                                                  GRID
           RLON = DEGTOR*RLON                                                  GRID
           RLATP = ABS(RLAT)                                                   GRID
           R = 124.8174804012*TAN((PI - 2.*RLATP)/4.)                          GRID
           A = RLON -PI/18.                                                    GRID
           FI = I0 + R*COS(A)                                                  GRID
           FJ = J0 - R*SIN(A)                                                  GRID
           IF(RLAT .LT. 0.)FI = FI + 2.*I0                                     GRID
           IGRID = FI  + .5                                                    GRID
           JGRID = FJ  + .5                                                    GRID
           IF(JSAVE .EQ. 0 .OR. JGRID .EQ. JSAVE)GO TO 40                      GRID
           CALL WRGRID(1)                                                      GRID
           CALL RDGRID                                                         GRID
        40 JSAVE = JGRID                                                       GRID
           IF(IREC(IGRID).EQ. 0)GO TO 50                                       GRID
           WRITE(IOUT,903) IGRID,JGRID                                         GRID
        50 IREC(IGRID) = IGRDPT                                                GRID
        70 CONTINUE                                                            GRID
           CALL WRGRID(-1)                                                     GRID
           GO TO 5                                                             GRID
       900 FORMAT(I5)                                                          GRID
       901 FORMAT(100,61H *** MINDEX IS NOT BETWEEN 0 AND 16000. PROGRAM IS TGRID
           1ERMINATED)                                                         GRID
```

```
000059          902 FORMAT(1H0,82H *** INDEX NUMBER READ FROM WEATAP IS NOT BETWEEN 1 GRID
000060             1AND 16000. THE DATA IS IGNORED)                                      GRID
000061          903 FORMAT(1H0,25H *** THE GRID PAIR IGRID=,I5,7H JGRID=,I5,37H HAS ANGRID
000062             1 INDEX NUMBER ALREADY ASSIGNED)                                      GRID
000063              END                                                                  GRID
```

```
* ELT RDGRID,1,760401, 68091    , 1

      SUBROUTINE RDGRID                                                 RDGRID
C     THIS ROUTINE READS 1 RECORD FROM DIRECT ACCESS FILE IGRDTP        RDGRID
      COMMON /FLAG/                                                     FLAG
     1INP,IOUT,TRTAPE,IGRDTP                                            FLAG
      COMMON /DIRAC/                                                    DIRAC
     1ICDCIN(252),IREC(500),IGRID,JGRID,JSAVE,MAXJ                      DEBUG1
      COMMON /WEATDT/                                                   WEATDT
     1IGRDPT,LAT(4),LON(4),IA(500),IB(50 ),IC(30)
      JG = JGRID + 1                                                    DEBUG1
      READ(IGRDTP'JG,ERR=10)(IREC(I),I=1,500)
      RETURN                                                            RDGRID
   10 WRITE(IOUT,901)
      STOP
  901 FORMAT(1H0,91H *** AN IRRECOVERABLE I/O ERROR HAS OCCURRED ON READ
     1ING A RECORD FROM THE INDEX MATRIX FILE /10X,31H THE JOB IS BEING
     2ABANDONED ***)
      END                                                               RDGRID
```

```
* ELT RDWETR,1,760401, 62093    , 1
                                                                        RDWETR
000001          SUBROUTINE RDWETR(IRC)                                  RDWETR
000002    C     THIS ROUTINE READS ONE PHYSICAL RECORD FROM NASA WEATHER TAPE  RDWETR
000003    C           NOT VALID ON UNIVAC                               DIRAC
000004          COMMON /DIRAC/                                          DEBUGI
000005         1IDCIN(252),IREC(500),IGRID,JGRID,JSAVE,MAXJ             FLAG
000006          COMMON /FLAG/                                           FLAG
000007         1INP,IOUT,IWTAPE,IGRDIP                                  WEATDT
000008          COMMON /WEATDT/
000009         1IGRDPT,LAT(4),LON(4),IA(500),IB(50 ),IC(30)             RDWETR
000010          DATA IC1/50/,IC2/50/                                    RDWETR
000011          IF(IRC.EQ. 1)GO TO 10
000012          CALL BTRAN(IWTAPE,2,500,IA,NSTAT,22)
000013        5 IF(NSTAT+1)500,5,10                                     RDWETR
000014       10 I =(IRC-1)*IC2+ 1
000015          DO 20 IG=1,IC2
000016          IB(IG) = IA(I+IG-1)                                     RDWETR
000017       20 CONTINUE
000018       50 DECODE(     900,IB(1))IGRDPT,(LAT(I),I=1,4),(LON(I),I=1,4)
000019          IRC = IRC + 1                                           RDWETR
000020          IF(IRC .GT. 10)IRC = 1                                  RDWETR
000021          RETURN                                                  RDWETR
000022      500 WRITE(IOUT,901)                                         RDWETR
000023          GO TO 515                                               RDWETR
000024      515 STOP                                                    RDWETR
000025      900 FORMAT(I5,13X,3I2,A1,1X,I3,2I2,A1)                      RDWETR
000026      901 FORMAT(1H0,52H*** END OF FILE OCCURRED ON WEATHER TAPE PREMATURELY
000027         1)                                                       RDWETR
000028          END                                                     RDWETR
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
      SUBROUTINE WRGRID(IOP)                                            WRGRID
C     THIS ROUTINE CREATES AND/OR WRITES RECORDS ON THE INDEX MATRIX    WRGRID
C     FTE - IGRDTP.                                                     WRGRID
      COMMON /HEATDT/                                                   HEATDT
     1IGRDPT,LAT(4),LON(4),IA(500),IB(50 ),IC(30)
      COMMON /DTRAC/                                                    DTRAC
     1ICDFIN(252),IREC(500),IGRID,JGRID,JSAVE,MAXJ                      DEBUG1
      COMMON /FLAG/                                                     FLAG
     1IOP,IOUT,IWTAPE,IGRDTP                                            FLAG
      IF(IOP)50,10,40                                                   WRGRID
   10 DEFINE FILE 4(42,500,U,IDUM)
      CALL NTRAN(IWTAPE,10,22)
      MAX =MAXJ - 1                                                     DEBUG1
      DO 20 I=1,MAX                                                     DEBUG1
      K = I
      FIND(IGRDTP'K)
      WRITE(IGRDTP'K,ERR=200)(IREC(L),L=1,500)
   20 CONTINUE                                                          WRGRID
      IREC(1) = 4HINDE                                                  DEBUG1
      IREC(2) = 4HXMAT                                                  DEBUG1
      FIND(IGRDTP'1)
      WRITE(IGRDTP'1,ERR=200)(IREC(I),I=1,500)
      IREC(1) = 0                                                       DEBUG1
      IREC(2) = 0                                                       DEBUG1
      GO TO 60                                                          WRGRID
   30 JG = JGRID + 1                                                    DEBUG1
      FIND(IGRDTP'JG)
      WRITE(IGRDTP'JG,ERR=200)(IREC(I),I=1,500)
C     CALL CLOSE(IGRDTP,0)
      CALL NTRAN(IWTAPE,11,22)
      GO TO 60                                                          WRGRID
   40 JS = JSAVE + 1                                                    DEBUG1
      FIND(IGRDTP'JS)
      WRITE(IGRDTP'JS,ERR=200)(IREC(I),I=1,500)
   60 RETURN                                                            WRGRID
  200 WRITE(IOUT,901)
      STOP
  901 FORMAT(1H0,91H *** AN IRRECOVERABLE I/O ERROR HAS OCCURRED ON READ
     1ING A RECORD FROM THE INDEX MATRIX FILE /10X,31H THE JOB IS BEING
     2ABANDONED ***)
      END                                                               WRGRID
```

7. T

LUMP BOOK III

Table of Contents

PART I

PROBLEM DESCRIPTION FOR LUMP PROGRAM

Problem Description for LUMP Program

1.0 SCOPE

1.1 PROGRAM CAPABILITIES

This program will take substrata historical data cards and sample segment ID, crop window, crop window error and statistical data cards, check the various fields for valid input parameters and write these cards as separate intermediate files on disk. It then reads these intermediate disk files and generates the Segment ID file, crop window file and the substrata historical file.

1.2 METHOD OF PROGRAM DEVELOPMENT

A main driver will do the organizing and will call routines FILGEN to generate the disk files and LUMP to generate the SEGMENT ID FILE, CROP WINDOW FILE and the SUBSTRATA HISTORICAL FILE. Existing utility routines will be used for angle conversion, data conversion and random number generation. Other existing routines will be utilized where deemed necessary.

1.3 OPERATIONAL ASSUMPTIONS

- Only 1 case is run at a time.
- Data Input Cards for the SAMPLE SEGMENTS are sorted by country, region, zone, strata, substrata, segment.
- Data Input Cards for the SUBSTRATA and CROP CALENDAR data are sorted by country, region, zone, strata, substrata.
- Program is a one pass process in which cards are checked, written to disk if no fatal errors, and then LUMP is executed.
- Maximum of 2000 training segments total.
- The input cards for SUBSTRATA HISTORICAL data, SUBSTRATA STATISTICAL data, CROP CALENDAR data, CROP CALENDAR ERROR data, and SAMPLE SEGMENT data are stored on separate input files.
- The United States must be assigned the symbol 'USA ' and Canada must be assigned the symbol 'CAND'.

2.0 INPUT

The only input media to this program is punched cards or card images on disk or tape files. They consist of a control card which includes the case number, random number seed, list option, training segment flag and unit numbers for the five possible types of input files. The other cards are the Substrata Historical Data, Substrata Statistical Data, Sample Segment ID data, the Sample Segment Crop Window data, and Crop Calendar Error data. (These last five sets of data may be card images on a disk or tape file.)

2.1 LIST OF INPUT QUANTITIES

See input data description sheets on Pages 141-143.

2.2 CARD FORMATS

See card format sheets on Pages 144 and 145.

2.3 DECK SETUP

The order of the input is:

1. Control Cards
2. Substrata Historical and Statistical Data Cards
3. Crop Window data cards and Crop Window Error data.
4. Sample segment ID cards.

2.4 RULES FOR ENTERING DATA ON CARDS

2.4.1 General

- Integers must be right justified.
- F-format or D-format numbers must have the decimal point present.
- There is a Crop Window card necessary for spring and winter. If either spring or winter are not required, only one card need be input.

2.4.2 Rules for Specific Fields

See Pages 4-6.

INPUT DATA DESCRIPTION

NAME	SYMBOL	DIMENSION	NOMINAL VALUE	RANGE	UNITS	DESCRIPTION
ICASE		1		0-9999		Identification case number for sample segment data (I5)
ISEG		1		0-9999		Segment number (I4)
ICTRY		1				Four character country name (i.e., USA, USSR)
IREG		1		0-999		Region number (I3)
IZONE		1		0-999		Zone number (I3)
ISTRAT		1		0-9999		Strata number (I4)
ISUB		1		0-9999		Substrata number (I4)
NAMSUB		2				Substrata name (2A4)
SUBAR				0-999999999	acres or hectares	Substrata total area (USA and Canada acres, USSR and others-hectares) (floating point, F10.0)
HISTPW	$\tilde{PW}$	1		0-100		% wheat in substrata (floating point, F7.3)
ISW		1		S, W		Spring or winter wheat indicator: S = Spring, W = Winter
DEVTPW	δPW	1		-9.999 to 9.999		Deviation to true proportion wheat (floating point, F6.3)
CV1	CV_1	1		0-9.999		Coefficient of variation for year-to-year change in proportion wheat (floating point, F5.3)
CV2	CV_2			0-9.999		Coefficient of variation for within county variation of proportion wheat (floating point, F5.3)
CV3	CV_3			0-9.999		Coefficient of variation for within county variation of proportion mixed pixels (floating point, F5.3)
DEVTPM	δPM			9.999		Deviation of true proportion mixed pixels (floating point, F5.3)
ITYPE				0-3		Card type (Ii) =0 Substrata Historical Data =1 Substrata Statistical Data =2 Substrata Crop Calendar Data =3 Substrata Crop Calendar Error Data =4 Sample Segment Location Data
LAT		1		$\pm$ 65°	Deg, Min	Latitude with N or S indicating + or - respectively (A1, I2, I3)

INPUT DATA DESCRIPTION (Cont.)

NAME	SYMBOL	DIMENSION	NOMINAL VALUE	RANGE	UNITS	DESCRIPTION
.ONG		1		±180°	Deg, Min	Longitude with E or W indicating + or 1 respectively (A1, I2, I3)
:MOT		8		1-12		Date month, day, year of each of 4 phases for start phase and stop phase 4(1X, 6I2) for winter or spring wheat
:DAT		8		1-31		
[YRT		8		64-99		
RSEED		1		1.0-NNNNNNNN NN		Floating point odd whole number used as seed to random number generator (D12.0)
[LIST		1		0.1		List option to list all input data cards or only those in error (I2) =0 Only those in error =1 List all input data cards
[SHD		1				Unit number of Substrata Historical Data (I2)
[SSD		1				Unit number for Substrata Statistical Data (I2)
[SID		1				Unit number for Segment ID Data (I2)
[SCW		1				Unit number for Segment Crop Window Data (I2)
[CWE						Unit number for Crop Calendar Error File (I2) - Set to 38 (8)
ITSFG				0, 1		Training Segment Flag (I2) =0 If all segments are training segments =1 If the training segment list is specified, via segment location data cards.
NAGR		1		1-9999		Number of agricultural segments in a substrata
NAL		1		1-9999		Number of allocated segments in a substrata
GV4		1		0-9.999		Multi-yr Std. deviation of historical wheat area
IGRP		1		1-3		Group no assignment for a substrata
DELTWE	$\delta_1-\delta_4$	4		±1.0		Delta error for winter in predicting the bio window dates, expressed at fraction of the window interval.
IPLNTE	δ_0	1		±99	Days	Error in the mean planting date of the substrata relative to the true mean value.

INPUT DATA DESCRIPTION (Cont.)

NAME	SYMBOL	DIMENSION	NOMINAL VALUE	RANGE	UNITS	DESCRIPTION
SEGSD	σ	1		0-99	Days	Within substrata variation for winter in crop calendar due to change in latitude, altitude, planting date, etc.
TRIND	-	1	0	0-1		0- training segment, 1- ordinary segment
TLIST	-	6		0-9999		List of segment numbers of training segments associated with the ordinary segment. These segments are to be listed in order of decreasing priority. Highest priority is first.
SW1	-	1		S or W		Spring or winter wheat indicator from
SW2	-	1		S or W		Crop calendar data
ELWE1	$\delta_1-\delta_4$	4		± 1.0		Delta error for spring - same as DELTWE
PLNT1	δ_0	1		± 99	Days	Same as IPLNTE only for spring
SGSD1	σ	1		0-99	Days	Same as for ISEGSD only for spring

CONTROL CARD

CC 1		9	21	26		31		36		41		46		51		56
Case No.	X	Random No. Seed	X	List Option	X	Training Seg Flag	X	Unit for Subst Hist	X	Unit for Subst Stat	X	Unit for Seg ID	X	Unit for Crop Calen.	X	Unit for Crop Error
I5		D12.0		I2		I2		I2		I2		I2		I2		I2

SUBSTRATA HISTORICAL DATA

CC 6	11	15	19	24	28	36	46	53	57	61	80
Country	Reg	Zone	Strata	Sub-Strata	Sub-Strata Name	Sub-Strata Total Area	Hist. PW	N_{AGR}	N_{AL}	Group	0
A4	I3	I3	I4	I4	A8	F10.0	F7.3	I4	I4	I1	

SUBSTRATA STATISTICAL DATA

CC 6	11	15	19	24	28	34	39	44	49	54	80
Country	Reg	Zone	Strata	Sub-Strata	Deviation True PW	Coeff. True PW	Coeff. True PW	Coeff. True PW	Ratio True PM	Multi-Yr Std. Dev. WA	1
A4	I3	I3	I4	I4	F6.3	F5.3	F5.3	F5.3	F5.3	F5.3	

SUBSTRATA CROP CALENDAR DATA

CC 6	11	15	19	24	29	42	55	68	80
Country	Reg	Zone	Strata	Sub-strata	S or W	Crop Phase 1	Crop Phase 2	Crop Phase 3	Crop Phase 4	2
						Start: Y M D; Stop: Y M D	Start: Y M D; Stop: Y M D	Start: Y M D; Stop: Y M D	Start: Y M D; Stop: Y M D	
A4	I3	I3	I4	I4	6I2	6I2	6I2	6I2		

1 Card For Spring and 1 For Winter

CROP CALENDAR ERROR DATA

CC 6	11	15	19	24	28	33	38	43	48	51	53	58	63	68	73	76	80
Country	Reg.	Zone	Strata	Sub-strata	BioWin 1 Error Winter δ_1	BioWin 2 Error Winter δ_2	BioWin 3 Error Winter δ_3	BioWin 4 Error Winter δ_4	Mean Plt. Date Error	Crop Cal. STD Dev.	BioWin 1 Error Spring δ_1	BioWin 2 Error Spring δ_2	BioWin 3 Error Spring δ_3	BioWin 4 Error Spring δ_4	Mean Plt. Date Error	Crop Cal. STD Dev.	3
A4	I3	I3	I4	I4	4F5.3				I3	I2	4F5.3				I3	I2	

SAMPLE SEG LOCATION DATA

1	6	11	15	19	24	29	36	44	46	48	80
	Country	Reg	Zone	Strata	Sub-strata	Latitude: N S, Deg, *, Min	Longitude: E W, Deg, *, Min	S or W	Tr. Index	Training Segment Priority List: Prior. 1, Prior. 2, Prior. 3, Prior. 4, Prior. 5, Prior. 6	4
	A4	I3	I3	I4	I4	A1I2 A1I2	A1I3 A1I2	A1	I1	6I4	

2.5 FILES

The only interface input file is INDMAT-index matrix location file. This file is used to get the index number associated with a segment latitude and longitude. See Section 2.4 of the Users Manual for format and contents.

3.0 PROCESSING

3.1 OVERVIEW

This program is arranged as a driver, an input processor, the main program (LUMP) and various utilities to accomplish their function: when called. The driver reads one data card and then calls the input processor which in turn performs checks on the various input fields as shown in Section 5.2. If no errors, the card image is written to an intermediate file and on option to the output file. If errors exist, the card is written to the output file along with the specified messages. The program will continue reading all the data cards and checking for errors but will terminate writing the intermediate card image file if the errors are fatal. If the errors are non-fatal, then the cards in error will not be written to the intermediate file and processing will continue when all cards have been checked. Additional runs may be made using this same data on the saved files, thus eliminating duplicate processing. After the intermediate files have been written, control is given to the main processor LUMP. Here one record of Substrata data (including a Crop Calendar Record) and one record of Segment ID data is read. The substratum number is checked to see if the segment belongs to this particular substratum. If not, the Substrata data is processed and written to the Substrata Historical file. Then another record of substratum data is read and processing continues. If the segment does belong to the substratum, the index number corresponding to the segment latitude and longitude is computed. The segment is then written to the Segment ID file, and another segment is read in and processed in the same manner until all segments for a particular substratum have been processed. A record is then written to the Crop Window File and the Substrata Historical File and new Substrata and Segment ID cards are read in and processed in the same manner until all cards have been read.

3.2 PROGRAM FLOW

Routine LUMPDR is the main driver which calls the input processor or the subroutine LUMP. Subroutine LUMP is the main body and performs all the needed calculations to generate the desired output files. An overview block diagram is presented on Page 151.

3.3 PROCEDURES AND EQUATIONS

3.3.1 Group No. Adjustment on Substrata Input Group No.

1. If a Substrata is typed a Group III a test is made to make sure there is no sample segment in this substrata. If there is, a message is printed and this situation is considered a fatal error.

2. If a substrata is typed a Group I, a test is made to make sure there is at least 1 sample segment in the substrata, if not a message is printed and the segment is retyped as a Group III.

3. If a substrata is typed a Group II, a test is made to see that there is at least one sample segment in one of the Group II substrata's within a governing strata.

3.3.2 Compute True Proportion of Wheat Using Beta Distribution.

1. Compute XBAR = HSTPW(1 + DEVTPW)/100.0
2. Compute SIGMA = HSTPW * CV1/100.00
3. Compute XI from subroutine BETAD given RSEED utilizing the incomplete beta function as computed via TRW canned routines. See subroutine BETAD writeup for details.
4. Compute TPWSS = XI * 100
5. If the subroutine BETAD can not find an answer for XI then TPWSS is SET = HSTPW.

HSTPW	-	Pseudo proportion of wheat
DEVTPW	-	Input deviation to true proportion of wheat
CV1	-	Input coefficient of variation
TPWSS	-	True proportion of wheat
RSEED	-	Random number seed
XBAR	-	Average proportion of wheat value
SIGMA	-	Standard deviation of proportion of wheat
XI	-	Random value for true proportion of wheat

3.3.3 Ensure Compatible Priority Training Segment Lists for Ordinary Segments.

A check is made to make certain that for each ordinary segment a valid list of priority training segments is given. This test is made only if the ITSFG flag is set for both ordinary and training input.

3.3.4 Compute predicted bio window start and stop dates and associated errors -t' for winter and spring:

$$t'_{o_1} = t_{o_1} + \delta_0$$

$$t'_{f_1} = t_{f_1} + \delta_1(t_{f_1} - t_{o_1}) + \delta_o$$

$$t'_{o_2} = t_{o_2} + \delta_1(t_{f_1} - t_{o_1}) + \delta_o$$

$$t'_{f_2} = t_{f_2} + \delta_1(t_{f_1} - t_{o_1}) + \delta_2(t_{f_2} - t_{o_2}) + \delta_o$$

$$t'_{o_3} = t_{o_3} + \delta_1(t_{f_1} - t_{o_1}) + \delta_2(t_{f_2} - t_{o_2}) + \delta_o$$

$$t'_{f_3} = t_{f_3} + \delta_1(t_{f_1} - t_{o_1}) + \delta_2(t_{f_2} - t_{o_2}) + \delta_3(t_{f_3} - t_{o_3}) + \delta_o$$

$$t'_{o_4} = t_{o_4} + \delta_1(t_{f_1} - t_{o_1}) + \delta_2(t_{f_2} - t_{o_2}) + \delta_3(t_{f_3} - t_{o_3})$$

$$+\delta_4(t_{f_4} - t_{o_4}) + \delta_o$$

$$t'_{f_4} = t_{f_4} + \delta_1(t_{f_1} - t_o) + \delta_2(t_{f_2} - t_{o_2}) + \delta_3(t_{f_3} - t_{o_3})$$

$$+ \delta_4(t_{f_4} - t_{o_4}) + \delta_o$$

Error Terms are:

$1 - \delta_o$	$3 - t'_{f_2} - t_{f_2}$	$5 - t'_{f_4} - t_{f_4}$
$2 - t'_{f_1} - t_{f_1}$	$4 - t'_{f_3} - t_{f_3}$	

$t_{o_1}-t_{f_4}$ - Reference crop calendar input start-stop dates

σ - Input planting date error

$\delta_1-\delta_4$ - Input error in predicting acquisition windows

3.3.5 Compute Weather Index No.

The program computes the grid I, J from the latitude and longitude of each segment and then pulls out the index number, from the input pattern matrix file INDMAT.

The equations and procedures are defined in the weather index generation document written by G. Rasmussen. The index pattern matrix will be generated by the utility 'GRID' from the NASA weather tape.

3.3.6 Compute PSEUDO proportion of wheat.

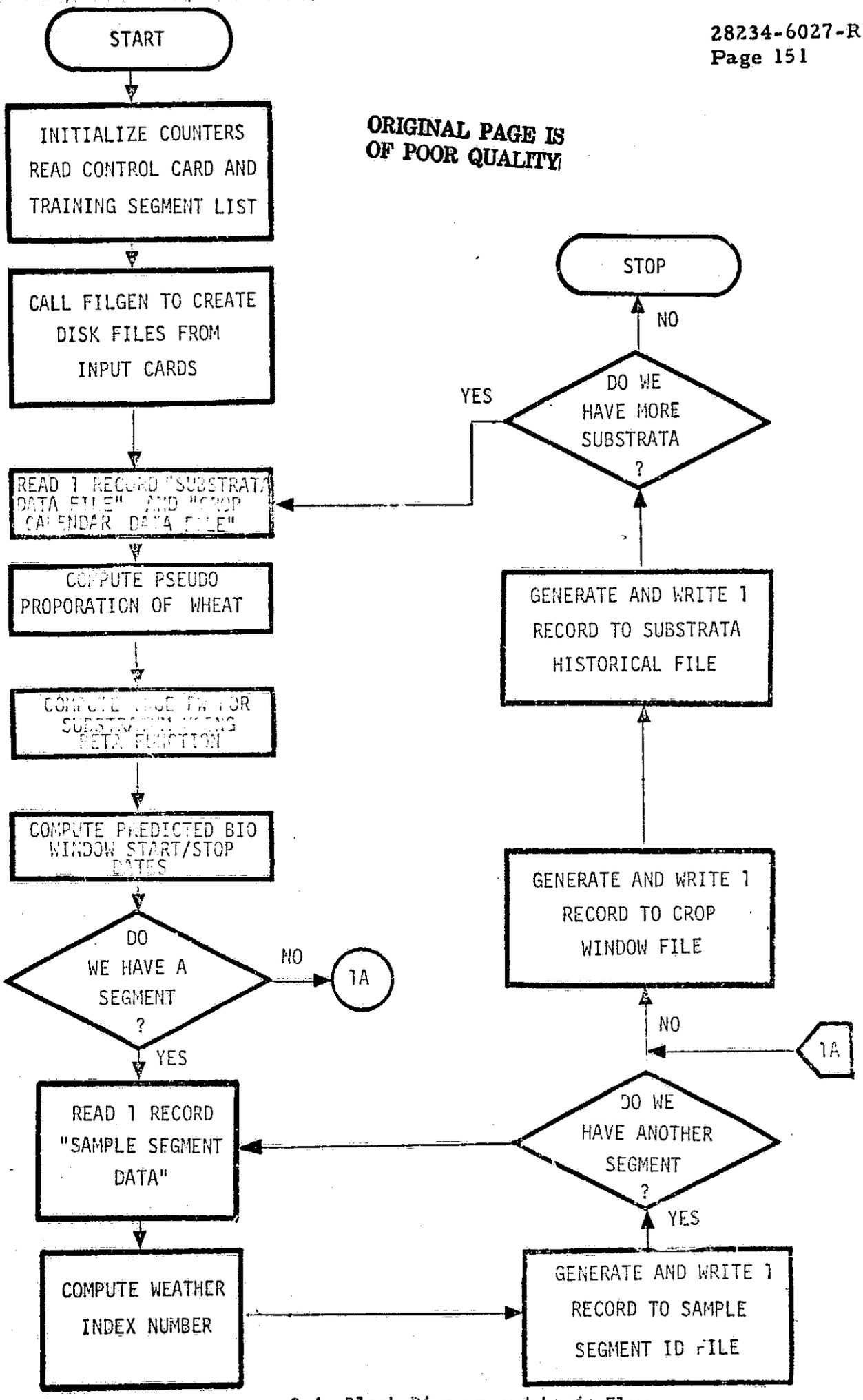
$$RK = \frac{AREA}{(NAL)*10289.712}$$

$$HSTPW = \frac{HISTPW*AREA}{NAGR*.RK*10289.712}$$

where:

AREA - Substrata land area in hectares
NAL - No of allocated segments in a substrata
HISTPW - Historical proportion of wheat in a substrata
NAGR - No of agricultural segments in a substrata

ORIGINAL PAGE IS
OF POOR QUALITY



3.4 Block Diagram and Logic Flow

4.0 OUTPUT

4.1 GENERAL

There are seven media for output. These are:

1. Print file
2. Sample Segment Intermediate File
3. Substrata Data Intermediate File
4. Crop Calendar Intermediate File
5. Segment ID File
6. Crop Window File
7. Substrata Historical File

The second and third are intermediate files and are images of the card input. The last three are the desired resultant of LUMP processing and are to be used later by other modules.

4.2 PRINTED OUTPUT

4.2.1 Standard Printout

The standard printout generated by the LUMP program consists of the following:

1. Listing of the quantities from the "control card"
2. Status recording
   - o A count of each type of card read
   - o Number of records written to each file
   - o A count of input errors
   - o A status flag indicating if job ran successfully
   - o Last random number seed used

4.2.2 Optional Printout

1. If ILIST $\geq$ 1 on the "control card", then the card images from all six input files (INPUT, Substrata Historical Data, Substrata Statistical Data, Sample Segment ID Data, Crop Window Data, and Crop Window Error Data are printed out.

2. If ILIST = 2, then the contents of the output files (segment ID file, Crop Window File, and Substrata Historical File) are printed out as each record is written (the records for the different files are intermixed on the printed output) although three separate files are actually generated.

4.3 DATA FILES

There are 3 output files: These are the Segment ID file, Substrata Historical file and Crop Calendar files. See Section 2.4 of the Users Manual for format and content.

5.0 ERROR PROCESSING

5.1 GENERAL

The program will attempt to find as many sources of error during input card processing as possible. The program will continue checking for further input errors upon detecting any input error. Processing will continue if possible if recovery from errors may be overcome.

5.2 INPUT DATA ERRORS

- Compatibility checks (Do the country, region zone, strata, and substrata agree between the substrata statistical data and between the sample segment ID data and the crop window data?)
- Checks for non-overlapping windows
- Sequence checking of the card data
- Input/output data limit checks
- Training priority segment NNNN is either not a training segment or nonexistant-fatal.
- A Group 1 substrata does not have at least 1 sample segment. Nonfatal
- A strata with at least 1 group 2 substrata does not have a sample segment in any group 2 substrata . Nonfatal
- A group 3 substrata has 1 or more sample segments . Fatal
- Region xxx zone xxx strata xxxx substrata xxxx beta distribution function could not converge on a solution
- Region xxx zone xxx strata xxxx substrata xxxx beta function reset ----- value since not between 0 and 1. Either ------ or ------ have bad values
- Region xxx zone xxx strata xxxx substrata xxxx segment xxxx. No crop calendar data available for segment intermediate file record, record skipped
- Region xxx zone xxx strata xxxx substrata xxxx segment xxxx. No weather grid index no. could be found. The segment record is skipped.

Input Data Limit Checks

Parameters	Range
Substrata Area	> 0
Historical PW	0 to 10^2
Deviation of True	-9.999 to 9.999
Yr-to-Yr CV of True PW	0 to 9.999
Within County CV of True PW	0 to 9.999
Within County CV of True PM	0 to 9.999
Multi Year CV of Standard Dev	0 to 9.999
Ratio of PM to PW	0 to 9.999
Spring/Winter Wheat Indicator	S or W
Segment Latitude	-65° to +65°
Crop Window Dates	
Year	64 - 99
Month	1 - 12
Day	1 - 31
Grid Index	1 - 16000
DELTWE, DELWE1	-1.0, +1.0

5.3 ERROR MESSAGES

Card Field Errors

For any error on a card the card image is printed prior to the message:

1. Inconsistent substrata historical and statistical data
2. Inconsistent Crop Window Error Data and Crop Window Data
3. Substrata area must be positive
4. Historical proportion of wheat must be non-negative
5. DEVTPW must be less than 9.999 in magnitude
6. COEFF, if variations must be between 0.0 and 9.999
7. DEVTPM must be between 0.0 and 9.999
8. Latitude must be less than or equal to 65.0 deg.
9. Longitude must be between -180 and 180 deg.
10. Spring/winter wheat indicator must be W or S
11. CV4 must be between 0.0 and 9.999
12. Year must be greater than 64
13. Month must be between 1 and 12
14. Day must be between 1 and 31
15. Overlapping crop window
16. Group no must be equal to 1, 2 or 3
17. Delta errors in bio window prediction dates are not between -1 and +1
18. Segment specified as ordinary must have a training segment priority list.
19. Neither NAGR or NAL can be zero.

PART II

COMMON BLOCK DEFINITIONS

COMMON STORAGE ALLOCATION

Name CONST Size 18 words Page 1 of 2

Function Program Constants

Name	Dimension	Format	Description	Symbol	Units
PI	1	F9.7	π = 3.1415926	π	
DEGTOR	1	F11.9	Degrees to radians conversion factor = π/180		radians/deg.
AREACF	1	F11.6	Area conversion factor = 4.046873E-3 KM^2/acre		KM^2/acre
			For U.S.A. and Canada = 0.01 KM^2/hectares otherwise		KM^2/hectare
SLATMX	1	F4.1	Minimum allowable latitude (65.0)		deg.
ENDATA	1	A4	End-of-data indicator (=4HZZZZ)		
ZERO	1	I2	Binary zero (0)		
INP	1	I2	Input file number (5)		
IOUT	1	I2	Output file number (6)		
INTSUB	1	I2	Substrata Intermediate File (30)		
INTSEG	1	I2	Segment Intermediate File (31)		
ISIDF	1	I2	Segment ID File number (1)		
ICWF	1	I2	Crop window file number (2)		
ISHF	1	I2	Substrata historical file number (3)		
LRSIDF	1	I2	Record length for the Segment ID File (13)		
LRCWF	1	I2	Record length for the Crop Window File (21)		
LRSHF	1	I2	Record length for the Substrata historical file (160)		
IWIN	1	I2	Number of crop windows (4)		
MXSEG	1	I2	Maximum number of segments in any substrata (150)		

COMMON STORAGE ALLOCATION

Name CONST Size _____ Page 2 of 2

Function Program Constraints

Name	Dimen-sion	For-mat	Description	Sym-bol	Units
ICALIN	1	I	Unit number of intermediate crop calender file		
ICWE	1	I	Unit number of crop calender error data		
ISW1	1	I	Spring or winter wheat input flag from crop calender cards		
ISW2	1	I	Spring or winter wheat input flag from crop calender cards		
IGRDTP	1	I	Unit number of index matrix file		

COMMON STORAGE ALLOCATION

Name FLAGS Size 182 Page 1 of 4

Function Flags and Indexes

Name	Dimen-sion	For-mat	Description	Sym-bol	Units
IERR	1	I2	Number of input errors		
IPRNT	1	I2	Print flag used in ERRMES ≠ 0 if the current card image has already		
			been printed = 0 otherwise		
GOOD1	1	I2	Error flag: ≠ 0 if the current substrata historical data		
			card or the current sample segment ID data card contains errors;		
			= 0 otherwise		
GOOD2	1	I2	Error flag: ≠ 0 if the current substrata statistical data		
			card or the current sample segment crop window data card contains errors;		
			= 0 otherwise		
NSHC	1	I2	Number of substrata historical data cards		
NSSC	1	I2	Number of substrata statistical data cards		
NSEGC	1	I2	Number of sample segment data cards		
NSUBR	1	I2	Number of records written on the substrata intermediate file		
NSEGR	1	I2	Number of records written on the segment intermediate file		
NRSIDF	1	I2	Number of records written on the segment ID File		
NRCWF	1	I2	Number of records written on the Crop Window File		
NRSHF	1	I2	Number of records written on the Substrata Historical File		

COMMON STORAGE ALLOCATION

Name FLAGS Size _____ Page 2 of 4

Function _____

Name	Dimen-sion	For-mat	Description	Sym-bol	Units
ISSC	1	I2	Substrata Statistical Data File read flag: ≠ 0 if the statistical data file was read		
			after reading the historical data file; = 0 otherwise		
ISTAT	1	I2	Substrata Statistical Data read flag: = 0 if for each substrata Historical Data		
			card, a statistical data card is to be read;		
			= 1 if the current statistical data card is to be used for all substrata within the		
			current strata; = 2 if the current statistical data card is to		
			be used for all strata within the current zone;		
			= 3 if the current statistical data card is to be used for sll zones within the		
			current region; = 4 if the current statistical data card is to		
			be used for all regions		
IREGL	1	I2	Previous region number		
IZONEL	1	I2	Previous zone number		
ISTRL	1	I2	Previous stratum number		
ISUBL	1	I2	Previous substratum number		
ISEGL	1	I2	Previous segment number		
NSEG	1	I2	Number of segments within the current substratum		
IMXSEG	1	I2	Maximum number of segments within any substrata		
NTLIST	1	I2	Number of segments in the training seg-ment ID list		

COMMON STORAGE ALLOCATION

Name FLAGS Size _____ Page 3 of 4

Function _____

Name	Dimen-sion	For-mat	Description	Sym-bol	Units
READSF	1	I2	Segment intermediate file read flag: ≠ 0 to skip reading of the segment inter-		
			mediate file until a new substratum is read on the Substrata Intermediate File; = 0 to read the segment intermediate file		
FATAL	1	I2	≠ 0 if a fatal error is detected in the input data		
IDAYS	8	I2	Array of crop window start/stop dates in Zulu date format:		
			IDAYS(2I-1) = start date IDAYS(2I) = stop date		
			for the I-th crop window for winter		
IDSEG	150	I2	List of segment ID numbers for a given substratum		

COMMON STORAGE ALLOCATION

Name FLAGS Size Page 4 of 4

Function

Name	Dimen-sion	For-mat	Description	Sym-bol	Units
IMOT	8		Temporary storage for crop calender stop/start dates Card 1 - Month		
IDAT	8		Card 1 - Day		
IYRT	8		Card 1 - Year		
IYR1	8		Card 2 - Year		
IMO1	8		Card 2 - Month		
IDA1	8		Card 2 - Day		

COMMON STORAGE ALLOCATION

Name INPUT Size 1071 Page 1 of 6

Function Input Quantities

Name	Dimen-sion	For-mat	Description	Sym-bol	Units
RSEED	1	F10.0	Random number seed		
SUBAR	1	F10.0	Substratum Total Area (acres for USA and Canada; hectares		acres or
			otherwise)		hectares
HISTPW	1	F7.0	Historical proportion of wheat (% wheat) converted to correct units for use in	$\widetilde{PW}$	
			subroutine LUMP		
DEVTPW	1	F6.3	Deviation of true proportion of wheat	δPW	
CV	3	F5.3	Coefficient of variation for year-to-year change in proportion of wheat	CV_1	
			Coefficient of variation for within county variation of proportion of wheat	CV_2	
			Coefficient of variation for within county variation of proportion of mixed pixels	CV_3	
DEVTPM	1	F5.3	Deviation of true proportion of mixed pixels	δPM	
QLAT	1	E10.6	Segment latitude		radians
QLONG	1	E10.6	Segment longitude		radians
SLAT	3		Segment latitude:		
		F4.0	SLAT(2) = degrees portion		
		F4.0	SLAT(3) = minutes portion		
SLONG	3		Segment longitude:		
		F4.0	SLONG(2) = degrees portion		
		F4.0	SLONG(3) = minutes portion		

COMMON STORAGE ALLOCATION

Name INPUT Size _____ Page 2 of 6

Function _____

Name	Dimension	Format	Description	Symbol	Units
NAMSUB	2	A4	Substrata name		
ICTRY	1	A4	Name of country on substrata Historical data card or on Sample Segment ID data card		
ICTRY2	1	A4	Name of country on Substrata Statistical Data card or on Sample Segment Crop Window Data card		
ICASE	1	I5	Case number		
ILIST	1	I2	List option = 0 for no printout except errors and statistics = 1 to printout input data = 2 to printout input data and the output files		
ITSFG	1	I2	Training segment flag ≠ 0 if the training segment list is specified = 0 if no training segment list is specified and if all segments are training segments		
ISHD	1	I2	Unit number for Substrata Historical Data		
ISSD	1	I2	Unit number for Substrata Statistical Data		
ISID	1	I2	Unit number for Sample Segment Statistical Data		
ISCW	1	I2	Unit number for Sample Segment Crop Window Data		
ISEG	1	I4	Segment ID (on Segment ID Data Card)		
IREG	1	I3	Region number		
ISZONE	1	I3	Zone number		
ISTRAT	1	I4	Strata Number		

COMMON STORAGE ALLOCATION

Name INPUT Size _____ Page 3 of 6

Function _____

Name	Dimen-sion	For-mat	Description	Sym-bol	Units
ISUB	1	I4	Substrata number Note: IREG, IZONE, ISTRAT, and ISUB		
			refer to the Substrata Historical Data or the Segment ID Data; IREG2, IZONE2, ISTRA2,		
			and ISUB2 refer to the Substrata statistical data or the segment crop window data.		
ISW	1	A1	Spring/winter wheat indicator: = S for spring wheat		
			= W for winter wheat		
ITYPE	1	I1	Card type = 0 for substrata hist. data		
			= 1 for substrata statistical data = 2 for segment ID data		
			= 3 for segment crop window data		
ITYPE2	1	I1	Card type - similar to ITYPE		
			ITYPE is used for the substrata historical data and the segment ID data and ITYPE2 is used		
			for the substrata statistical data and the crop window data		
INDEX	1	I5	Segment index (grid number)		
ISEG2	1	I4	Segment ID (on crop window data card)		
IREG2	1	I3	Region number		
IZONE2	1	I3	Zone number		
ISTRA2	1	I4	Strata number		
ISUB2	1	I4	Substrata number Note: See the note following description of		
			ISUB above		

COMMON STORAGE ALLOCATION

Name INPUT Size _____ Page 4 of 6

Function _____

Name	Dimen-sion	For-mat	Description	Sym-bol	Units
LAT	3	A1	Segment latitude		
		I2	LAT(1) is N or S LAT(2) is the degrees portion		degrees
		I2	LAT(3) is the minutes portion		minutes
LONG	3	A1	Segment longitude		
		I2	LONG(1) is E or W LONG(2) is the degrees portion		degrees
		I2	LONG(3) is the minutes portion		minutes
IYR	8	I2	Year portion of the start/stop dates (≥ 64)		years
IMO	8	I2	month portion of the start/stop dates (1 - 12)		months
IDA	8	I2	day portion of the start/stop dates (1-31)		days
			IYR(2n-1) is the start date of the nth crop window		
			IYR(2n) is the stop date of the nth crop window		

COMMON STORAGE ALLOCATION

Name INPUT Size _____ Page 5 of 6

Function _____

Name	Dimension	Format	Description	Symbol	Units
IGRP	1	I	Substrata group number		
CV4	1	R	Multi year standard deviation		
DELTWE	4	R	Bio window error set for winter	δ_i	
IPLNTE	1	I	Mean planting date error for winter	δ_o	
ISEGSD	1	I	Crop calender standard deviation for winter		
ITRIND	1	I	Training segment indicator		
NAGR	1	I	Number of agricultural segments in a substrata		
NAL	1	I	Number of allocated segments in a substrata		
ITLIST	6	I	List of priority training segments		
NCRCD	1		Number of crop calender cards read		
NCRECD	1		Number of crop calender error cards read		
NCROP	1		Number of records written on crop calender file		
IDAYS1	8		Crop calender start/stop zulu dates for spring		
DELWE1	4		Bio window error set for spring		
IPLNT1	1		Mean planting date error for spring		
ISGSD1	1		Crop calender standard deviation for spring		
IDAS	8		Input crop calender start stop set for spring day set		
IMOS	8		month set		

COMMON STORAGE ALLOCATION

Name INPUT Size _____ Page 6 of 6

Function _____

Name	Dimen-sion	For-mat	Description	Sym-bol	Units
IYRS	8	I	Year set		

COMMON STORAGE ALLOCATION

Name LDATA Size 16 Page 1 of 1

Function Spring/Winter Wheat Crop Window Dates

Name	Dimen-sion	For-mat	Description	Sym-bol	Units
WWDATA	8	I5	Winter wheat crop window start/stop dates (Zulu Dates - no. of days since 1 Jan 1950)		
SWDATA	8	I5	Spring wheat crop window start/stop dates		
			(Zulu Dates) WWDATA(2n-1) is the start date		
			WWDATA(2n) is the stop date for the nth crop window		
			Similarly for SWDATA		

COMMON STORAGE ALLOCATION

Name CALEND Size _____ Page 1 of 1

Function _____

Name	Dimen-sion	For-mat	Description	Sym-bol	Units
IPREDW	8	I	Predicted start and stop date set for winter	t'_{o_i}, t'_{f_i}	
IPREDS	8	I	Predicted start and stop date set for spring	t'_{o_i}, t'_{f_i}	
ICRPEW	8	I	Crop window error set for winter		
ICRPES	8	I	Crop window error set for spring		

COMMON STORAGE ALLOCATION

Name TRATAB Size _____ Page 1 of 1

Function Training Table Data

Name	Dimen-sion	For-mat	Description	Sym-bol	Units
ITRAIN	2000	I	List of all training segments specified via sample segment location data		
ITRPRI	2000	I	List of unique segment IDs identified in priority lists for ordinary segments		

COMMON STORAGE ALLOCATION

Name DIRAC Size _____ Page 1 of 1

Function Direct Access Storage Index Matrix File

Name	Dimen-sion	For-mat	Description	Sym-bol	Units
ICDIN	252	I	Array for index record pointers for CDC only		
IREC	500	I	Contains 1 physical record from the index matrix file		

CROP CALENDAR

Intermediate Data File - CRPINT

This file contains the substrata crop calendar and its corresponding error data stored in one record. This file is used by subroutine LUMP.

Access Method: Sequential

Status: Temporary

Sort: Country through substrata

Media: Drum

Record Format:

Detail Record

ICTRY - Country ID - A4
IREG - Region no.
IZONE - Zone no.
ISTRAT - Strata no.
ISUB - Substrata
IDAYS - Four stop/start Zulu date pairs for winter
DELTWE - Four values of bio-window errors for winter
IPLNTE - Winter mean planting date error
ISEGSD - Winter crop calendar standard deviation
IDAYS1 - Four stop/start Zulu date pairs for spring
DELWE1 - Four values of bio-window errors for spring
IPLNT1 - Spring mean planting date error
ISGSD1 - Spring crop calendar standard deviation
Total - 33 words

Trailer Record

First word has 'ZZZZ' followed by 32 words of zero

Recommended Blocking Factor: 10

File Size: 105,633 words max.

SUBSTRATA DATA

Intermediate File - SUBINT

This file contains the combined data from the substrata historical and statistical data card sets. It is used by subroutine LUMP.

Access Method: Sequential

Status: Temporary

Sort: Country through substrata

Media: Drum

Record Format:

Detail Record

ICTRY	-	Country ID - A4
IREG	-	Region no.
IZONE	-	Zone number
ISTRAT	-	Strata no.
ISUB	-	Substrata
NAMSUB	-	Two word substrata name - 2A4
AREA	-	Substrata area
HSTPW	-	Historical proportion of wheat converted to proper units
DEVTPW	-	Deviation of true proportion of wheat
CV	-	Three coefficients of variation
DEVTPM	-	Deviation of true proportion of mixed pixels
NAGR	-	No. of agricultural segments
NAL	-	No. of allocated segments
CV4	-	Multi-year standard deviation
IGRP	-	Group no.

Total - 18 words

Trailer Record:

First word has 'ZZZZ' followed by 17 words of 0

Recommended Blocking Factor: 10

File Size: 57,618 words max.

SEGMENT DATA

Intermediate File - SEGINT

This file contains the data from the sample segment cards. It is used by subroutine LUMP.

Access Method: Sequential

Status: Temporary

Sort: Country through segment

Media: Drum

Record Format:

Detail Record:

ICTRY	-	Country ID - A4
ISEG	-	Segment no.
IREG	-	Region no.
IZONE	-	Zone no.
ISTRAT	-	Strata no.
ISUB	-	Substrata no.
QLAT	-	Latitude
QLONG	-	Longitude
ISW	-	Spring/winter flag
ITRIND	-	Training segment flag
ITLIST	-	Six training segment ID's

Total - 16 words

Trailer Record:

First word has 'ZZZZ' followed by 15 words of 0

Recommended Blocking Factor: 10

File Size: 64,000 words max.

PART III

LIST OF SUBROUTINES
AND
SUBROUTINE CALL STRUCTURE

LIST OF ROUTINES IN LUMP

LUMPDR	Main control routine for the LUMP program.
FILGEN	Reads data from the four input files and generates the Substrata Intermediate File and the Segment Intermediate File.
LUMP	Reads data from the Substrata Intermediate File and from the Segment Intermediate File and generates the Segment ID file, the Crop Window file, and the Substrata Historical file.
ERRMES	Error Message routine. Prints most of the error messages.
LFPA	Calculates the Zulu Date (Number of days from 1 January 1950) from a given calendar date (year, month, day).
RDMIA	Random number generator.
CALGEN	Reads, checks and merges crop calender and crop calender error data. Writes intermediate file.
CALERR	Computes predicted bio window start and stop dates with their associated errors.
BETAD	Compute a random number based on the beta distribution or normal distribution.
IBETAI	Computes incomplete beta function (TRW utility)
ALGAMA	Computes gamma function (TRW utility)
GRID	Controls the calculation of the weather grid index number, given latitude and longitude
GRIDCAL	Computes the best index value for a given grid pair of coord. in the global stereographic index matrix.
RDGRID	Opens or closes IGRDTP file or reads a record from it (the Index Matrix File).

SUBROUTINE CALL STRUCTURE

```
LUMPDR
    RDGRID
    FILGEN
        ERRMES
        CALGEN
            ERRMES
        LFPA
    LUMP
        CALERR
        BETAD
            RDMIA
            IBETAI
                ALGAMA
        GRID
            GRDCAL
                RDGRID
```

PART IV

SUBROUTINE DESCRIPTIONS AND FLOWCHARTS

SUPPLIED UTILITY ROUTINES

Routine Day

Call Day (IYMD, IDAY)

Given IYMD (3) where

IYMD (1)	IS Day No.
IYMD (2)	IS Month No.
IYMD (3)	IS Year No.

Compute year day no. in IDAY

Routine PIMOD

Call PIMOD (A)

Convert ±A in radians to an angle 0-2π

Routine SOL (Entry ALPHA)

Call ALPHA (IFLAG)

For emphemeris usage as called by hector

computes ALPHAM and ALPHAT and IFLAG = 1

Routine PAGER (Entry Eject)

Call PAGER (NLINES)

Updates line count in NLINE with NLINES

NPAGE = 0 causes page to be restored prior to print.

NPAGE - page no.

HEADER - 80 char. 20A5

ICASE - case no.

KO - 6 print unit

INMAX is max no. of lines allowed

Initially NLINE should be set > LINMAX and NPAGE = 0

SUPPLIED UTILITY ROUTINES
(CONTINUED)

Call EJECT (NLINES)

Causes page to be restored automatically and then prints headers.

Routine CLDAY

Call CLDAY

Given IDAY-DAY no. of the year compute in LMO-the month and in LDA the day no.

Need: IYEAR = 0 - Leap Year, ≠ 0 not Leap Year

Routine KEPLER

Call KEPLER (XM, XECC, XE, ERROR)

Given XM - Mean anomaly, XECC - eccentricity

Compute: E-eccentric anomaly, error = 0 means OK

Routine LFPA

Call LFPA [FLDA, LMO, LYR, ALFGM (can be dummy), DAYS]

Given: FLDA - day of month no., LMO - month no.,
LYR - year no. compute ALFGM - right ascension and
DAYS - Zulu day no.

Routine DEGMOD

Call DEGMOND (RAD, IDEG)

Given: angle rad in radians store the angle in deg., min., sec., in IDEG(1) - (3).

Routine FZULU

Call FZULU (IOATE, IOUT)

Given Zulu date in IDATE, compute year, month and day in IOUT(1) - IOUT(3).

Routine RDMLA

Call RDMLA(FL, U)

Given double precision random no. seed in FL, comoute random no. U (0-1) based on uniform distribution.

Subroutine LUMPDR

Purpose: Main control routine for the LUMP program

Linkage: Entered from the Operating System

Inputs:

The input quantities for LUMPDP are read from the INPUT file and consist of all the information on the "Control" card plus the Training segment ID list.

Outputs:

The output quantities from LUMPDP are the quantities read from the INPUT file and are stored in the labeled COMMON block /INPUT/ for use by other routines. In addition the quantity NTLIST, the number of entries in the Training Segment ID list is computed and stored in the COMMON block /FLAGS/.

Subroutines Called:

FILGEN - reads data from the four input files and generates the Substrata Intermediate File and the Segment Intermediate File.

LUMP - reads data from the two intermediate files and generates the Segment ID File, the Crop Window File, and the Substrata Historical File.

RDGRID - read and check header record on IND mat file

Error Messages:

None

Definitions of Local Variables:

I Index in DO LOOP to read Training Segment ID list.

J Index in DO LOOPs to read Training Segment ID list.

L1 Lower limit of J in the DO LOOP for a given card image.

L2 Upper limit of J in the DO LOOP for a given card image.

Subroutine FILGEN

Purpose: Reads data from three input files and generates the Substrata Intermediate Files and the Segment Intermediate File.

Linkage: CALL FILGEN
There are no arguments in the calling sequence. All input/ourput quantities are transmitted through COMMON storage or disk/tape/drum files.

Input:

Most of the input quantities are specified on three input files

1. Substrata Historical Data
2. Substrata Statistical Data
3. Sample Segment ID Data

In addition, the following quantities are required and are specified through COMMON storage:

ILIST	List option: = 1 to list input data; = 0 otherwise
ISHD	File number for Substrata Historical Data (=0 if the Substrata Intermediate File from a previous run is to be used)
ISSD	File number for Substrata Statistical Data (=0 if the Substrata Intermediate File from a previous run is to be used)
IOUT	File number for the print file on which card images, error messages, and run statistics may be printed
ISID	File number for Sample Segment ID Data (= 0 if the Segment Intermediate File from a previous run is to be used)

ENDATA	End of data indicator (= 4HZZZZ)
DEGTOR	Degrees to radians conversion factor (= π /180)
INTSUB	File number for Substrata Intermediate File
INTSEG	File number for Segment Intermediate File

Output:

The output data generated by FILGEN is written on the Substrata Intermediate File and the Segment Intermediate File. In addition, the following statistical quantities are computed and printed out:

NSHC	Number of Substrata Historical Data cards read
NSSC	Number of Substrata Statistical Data Cards read
NSEGC	Number of Sample Segment ID Data cards read (also the number of Sample Segment Crop Window Data cards)
NSUBR	Number of records written on the Substrata Intermediate File
NSEGR	Number of records written on the Segment Intermediate File
IERR	Number of errors detected on the input data

Subroutines Called:

ERRMES	Error message subroutine
CALGEN	Reads crop calender data and writes intermediate file

Error Messages:

The following error codes may be used. See the writeup of subroutine ERRMES for the complete error message:

3, 4, 5, 6, 7, 8, 9, 10, 11, 16, 18, 19

Definitions of Local Variables:

AREA	Area of substratum in Km^2
DAYS	Zulu date (in floating point)
FLDA	Day portion of calendar date
HSTPW	Historical proportion of wheat (true percent)
I	Index in miscellaneous DO LOOPs.
ISKCW	Flag controlling reading of Sample Segment Crop Window Data: = 0 to read one card image from the Crop Window Data after reading one card image from the Segment ID Data = 1 to skip reading the Crop Window Data temporarily until the Segment ID Data catches up with the Crop Window Data (i.e., until the segment IDs match)
ISKSSC	Flag controlling reading of the Substrata Statistical Data: = 1 to skip reading of the Substrata Statistical Data until the Substrata Historical Data catches up with the Statistical Data (i.e., until the region, zone, strata, and substrata agree unless the region, zone, strata, or substrata from the statistical data are zero)
ICT	Count of number of training segments associated with and ordinary segment specified in a priority list.
ILOC	Current location in ITRAIN array
ILOCI	Current location in ITRPRI array
ISEGPR	Current priority training segment ID used to make sure segment input data has specified valid input priority training segment lists.

Subroutine LUMP

Purpose: Reads data from the Substrata Intermediate File Crop Calender Intermediate File and the Segment Intermediate File and generates the Segment ID file, the Crop Window File, and the Substrata Historical File.

Linkage: CALL LUMP
There are no arguments in the calling sequence. All input/output quantities are transmitted through COMMON storage or disk/tape/drum files.

Input:

Most of the input quantities are specified on the Substrata Intermediate File and the Segment Intermediate File. In addition, the following quantities are required and are specified through COMMON storage:

INTSUB	File number for the Substrata Intermediate File
INTSEG	File number for the Segment Intermediate File
ISHF	File number for the Substrata Historical File
ISIDF	File number for the Segment ID File
ICWF	File number for the Crop Window File
LRSHF	Length of one record of the Substrata Historical File
LRSIDF	Length of one record of the Segment ID File
LRCWF	Length of one record of the Crop Window File
ICASE	Case number
RSEED	Random number seed
ICALIN	File number of crop calender intermediate file

ILIST	List option = 1 to print out input card images (in subroutine FILGEN) = 2 to print out the contents of the three output files = 0 to suppress all printing except for error messages and statistics
ITSFG	Training Segment option = 1 if the Training Segment ID list is used; = 0 if all segments are training segments
IOUT	File number for print file
IWIN	Number of crop windows (4)
MXSEG	Maximum number of segments in any substrata (150)
ENDATA	End-of-data indicator (4HZZZZ)
ZERO	Zero constant

Output:

The output data is written on the following three files:

1. Segment ID File
2. Crop Window File
3. Substrata Historical File

Subroutines Called:

CALERR	Computes predicted bio window start/stop dates
BETAD	Generate random number using incomplete beta distribution
GRID	Computes the weather index for segments latitude and longitude

Error Messages:

1. "THE FOLLOWING SEGMENT DATA WAS DROPPED DUE TO DISAGREEMENT WITH THE SUBSTRATA DATA." followed by a printout of the segment, country, region, zone, stratum, and

substratum from the current record of segment data from the Segment Intermediate file and from the current record of substrata data from the Substrata File. This error message indicates that the indicated set of Segment Data cannot be matched with any substrata on the Substrata Intermediate File. Possibly one file or the other is out of order.

2. "THE FOLLOWING SEGMENT DATA WAS DROPPED BECAUSE IT DID NOT BELONG TO ANY STRATA ON THE SUBSTRATA HISTORICAL FILE" followed by a printout of the segment, country, region, zone, stratum, and substratum from the current record of the Segment Intermediate File. This message indicates that the indicated segment is being dropped because there are no more data records available on the Substrata Intermediate File. Hence there is no way to match up the indicated segment with the proper set of Substrata Data. Again one file or the other may be out of order.

3. "REGION NNN ZONE NNN STRATA NNNN SUBS NNNN BETA FUNCTION COULD NOT CONVERGE ON A SOLUTION." This message means that subroutine BETAD could not on a solution for RSEED.

4. "REGION NNN ZONE NNN STRATA NNNN SUBSTRATA NNNN SEGMENT NNNN NO CROP CALENDAR DATA AVAILABLE FOR SEGMENT INTERMEDIATE FILE RECORD. SEGMENT RECORD SKIPPED." Self explanatory. The ID is for the segment record being skipped.

5. "REGION NNN ZONE NNN STRATA NNNN SUBSTRATA NNNN A GROUP 3 SUBSTRATA HAS 1 OR MORE SAMPLE SEGMENTS." A group 3 substrata must not have any sample segments. This error is job fatal.

6. "REGION NNN ZONE NNN STRATA NNNN SUBSTRATA NNNN A GROUP 1 SUBSTRATA DOES NOT HAVE AT LEAST 1 SAMPLE SEGMENT." A group 1 substrata is supposed to have 1 or more sample segments. The group number is reset to 3.

7. "REGION NNN ZONE NNN STRATA NNNN A STRATA WITH AT LEAST 1 GROUP 2 SUBSTRATA DOES NOT HAVE A SAMPLE SEGMENT IN ANY GROUP 2 SUBSTRATA." A group 2 strata must have at least 1 sample segment.

8. "REGION NNN ZONE NN STRATA NNNN SUBSTRATA NNNN BETA FUNCTION RESET AAAAAA VALUE SINCE NOT BETWEEN 0 AND 1.
EITHER AAAAAA OR AAAAAA HAVE BAD VALUES, NNN.NNN NNN.NNN." Subroutine BETAD can not operate outside the specified limits.

Definitions of Local Variables:

AREA	Area of a substratum in Km^2
CWHDR	Two word array containing the header for the Crop Window File in the format 4HCROP, 4HWIND
I	Index in DO LOOPs
ISEGR	Index for number of records read from the Segment Intermediate File
ISTRTP	Strata type flag: = 1 if there is at least one segment in the current substratum; = 0 otherwise
NFILL	Number of words required to complete the header record or the trailer record of one of the output files

SHHDR — Two word array containing the header for the Substratum Historical File in the format
4HSUB, 4HHIST

SIDHDR — Two word array containing the header for the Segment ID File in the format
4HSEGM, 4HENT

TPWSS — PW_K = Temporary location for * proportion wheat for a substratum

IREGP, IZONP, ISTRP - Previous substrata history intermediate file ID (Reg. - Strata)

ICRPR — ≠ 0 Do not read crop calendar intermediate file
= 0 Okay to read

ICRPPR — ≠ 0 Crop calendar ID does not match substrata history intermediate file
= 0 IDs match

ISSUB — Number of segments in a substrata (used in group processing logic)

IGRP2 — Number of Group 2 segments in a strata

ICV1 — Constant 'CV1'

IDEVTP — Constant 'DEVTPW'

IHISTP — Constant 'HISTPW'

ISIGMA — Constant 'SIGMA'

ICTRY1, IREG1, IZON1, ISTRA1, ISUB1 - Crop calendar intermediate file ID

IGRP1 — Number of Group 1 segments in a strata

ISWH — Flag indicating whether segment data is spring = 1 or winter = 0

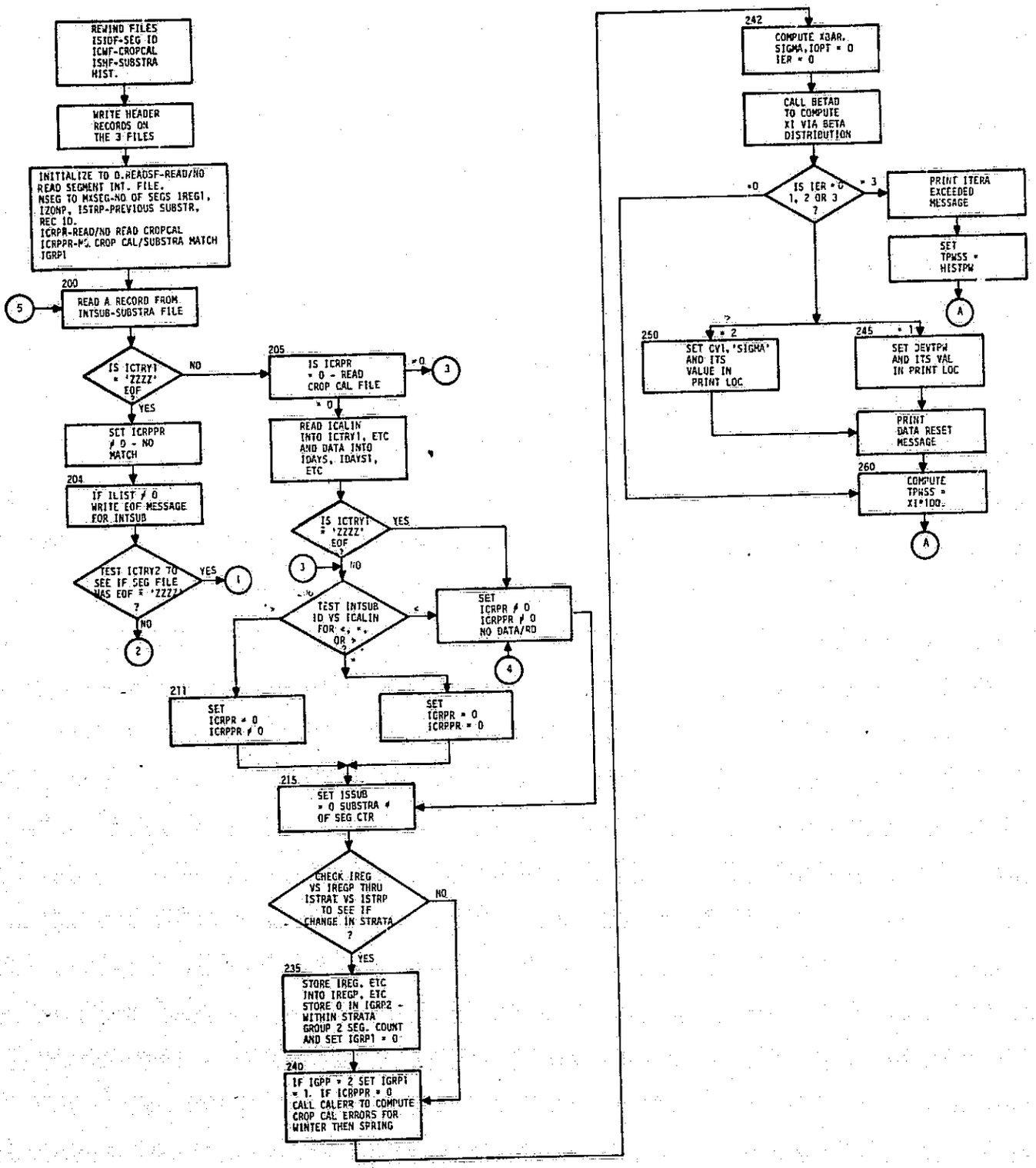
XI — Random number output from BETAD

XBAR — Mean value of variable input to BETAD

SIGMA — Standard deviation of variable

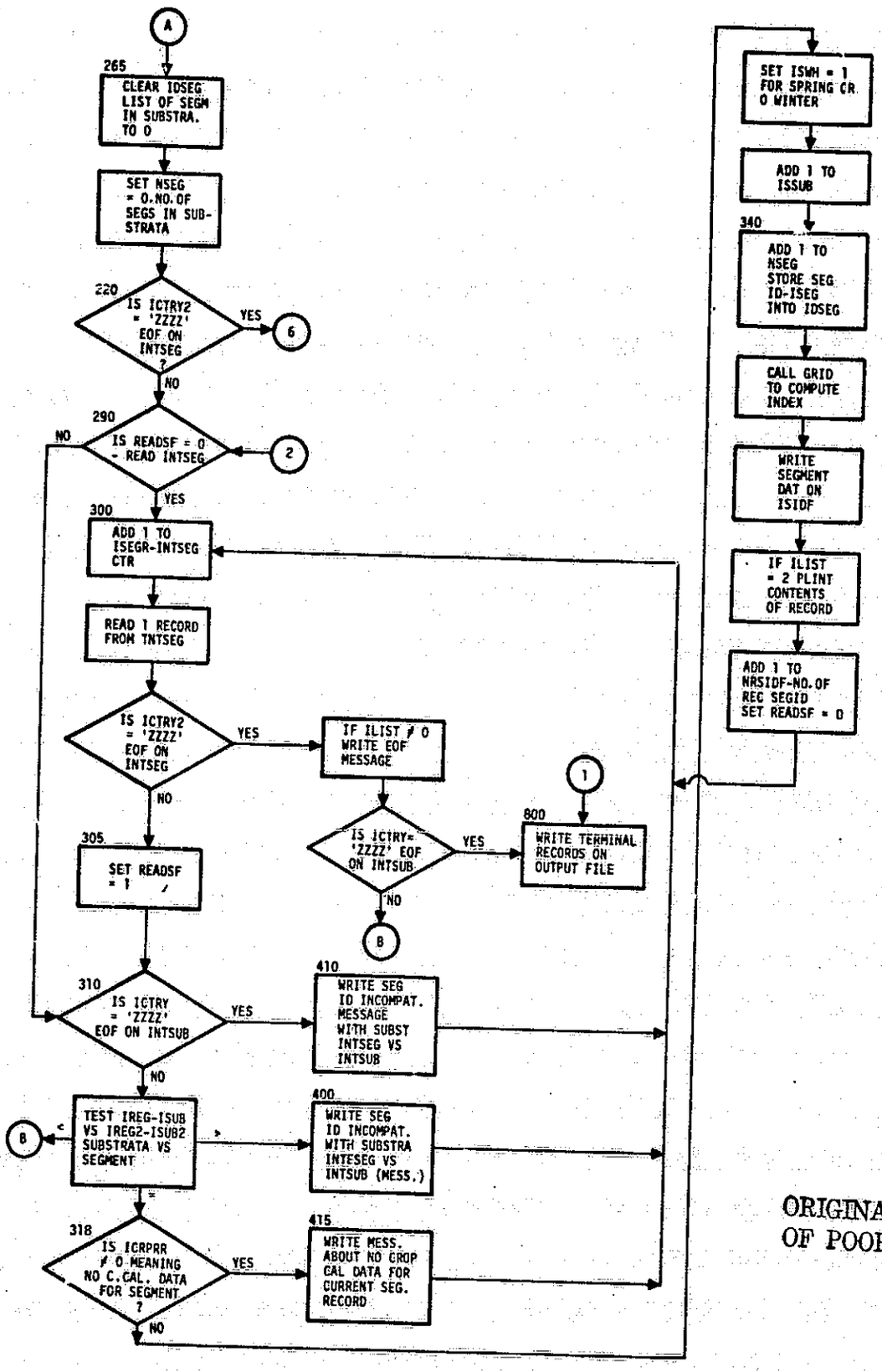
C.3

LUMP FLOWCHART

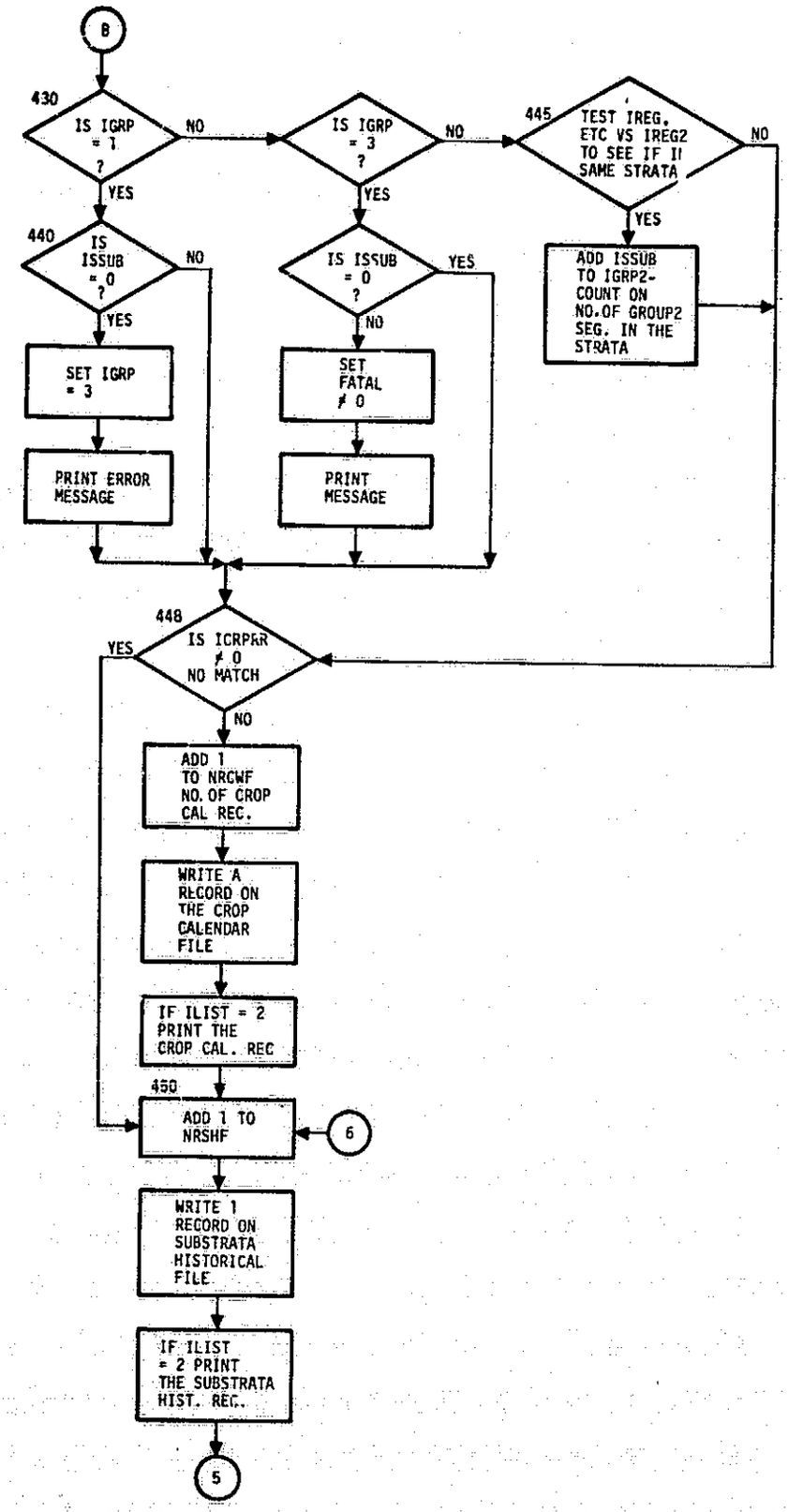


ORIGINAL PAGE IS
OF POOR QUALITY

LUMP (CONTINUED)



LUMP FLOWCHART



SUBROUTINE ERRMES

Purpose: Error message subroutine: Writes error messages on the print file as directed by flags in the calling sequence.

Linkage: Call ERRMES (ICARD, ICODE)

Input: ICARD - Type of card containing the error.
= 1 for substrata historical data;
= 2 for substrata statistical data;
= 3 for sample segment ID data;
= 4 for substrata crop widow data.
= 5 for crop window error data.

ICODE - Error code (1 - 19)

In addition to the above two quantities specified through the argument list, the following input quantities are also required and are transmitted through labeled common:

IPRNT - Card image print flag:
= 0 if the current pair of card images have not been printed;
≠ 0 if they have been printed.

IERR - Previous number of input errors.

ICTRY IREG IZONE ISTRAT ISUB ISEG	The country, region, zone, stratum, substratum and segment from either the Substrata Historical Data, the Sample Segment Data or substrata crop window data.
ICTRY2 IREG2 IZONE2 ISTRA2 ISUB2 ISEG2	The country, region, zone, stratum, substratum and segment from either the Substrata Statistical Data or the Crop Window Error Data.

NAMSUB SUBAR HISTPW ITYPE NAGR NAL IGRP	The substrata name, area, Historical PW, etc., and card type from the Substrata Historical Data.
DEVTPW CV DEVTPM CV4 ITYPE2	The Deviation in the True PW, the coefficients for the True PW, and the ratio of the True PM and the card type from the Substrata Statistical Data.
LAT LONG ISW ITRIND ITLIST ITYPE	The latitude longitude, spring/winter wheat indicator, etc., and the card type from segment ID data
ISW1 IYR IMO IDA ISW2 IYR1 IMO1 IDA1	Origins of three arrays dimensioned 8 in which the crop windows are stored by year, month, and day. For spring and winter, from substrata crop window data
DELTWE IPLNTE ISEGSD DELWE1 IPLNT1 ISGSD1	Data from crop calendar error data for spring and winter.

Output: IPRNT - Card image print flag set = 1 if not already 1 to indicate that the pair of card images were printed out.

GOOD1 - Flag set = 1 to indicate that the Substrata Historical Data or the sample segment ID data is no good.

GOOD2 - Flag set = 1 to indicate that the Substrata Statistical Data or the Sample Segment Crop Window Data is no good.

IERR - The error count is advanced by 1 each time ERRMES is called.

FATAL- Set = 1 if there are inconsistencies between the Substrata Historical Data and the Substrata Statistical Data or between the Sample Segment ID Data and the Sample Segment Crop Window Data.

Subroutines Called: None

Error Messages: The type of card containing the error is printed out followed by the card data. Then one of the following error messages is printed.

1. INCONSISTENT SUBSTRATA HISTORICAL AND STATISTICAL DATA - The country, region, zone, strata, and/or substrata fields do not agree on the indicated cards. Possibly, cards are out of order or missing on one of the input files.

2. INCONSISTENT CROP WINDOW AND CROP WINDOW ERROR DATA - The segment, country, region, zone, strata, and/or substrata fields do not agree on the indicated cards. Possibly, cards are out of order or missing on one of the input files.

3. SUBSTRATA AREA MUST BE POSITIVE - The substrata area is negative or zero or unspecified. It must be > 0.

4. HISTORICAL PROPORTION OF WHEAT MUST BE NON-NEGATIVE - HISTPW is negative. It must be ≥ 0.

5. DEVTPW MUST BE LESS THAN 9.999 IN MAGNITUDE - DEVTPW is > 9.999.

6. COEFF. OF VARIATIONS MUST BE BETWEEN 0.0 AND 9.999 - One of the quantities CV_i is < 0 or > 9.999.

7. DEVTPM MUST BE BETWEEN 0.0 AND 9.999 - DEVTPM < 0 or > 9.999.

8. LATITUDE MUST BE LESS THAN OR EQUAL TO 65.0 DEG - The magnitude of the segment latitude is greater than SLATMX, the maximum allowable latitude.

9. LONGITUDE MUST BE BETWEEN -180 AND +180 DEG. - The magnitude of the segment longitude is greater than 180°.

10. SPRING/WINTER WHEAT INDICATOR MUST BE W OR S - The quantity ISW is unspecified or is a character other than W or S.

11. CV4 MUST BE BETWEEN 0.0 AND 9.999.

12. YEAR MUST BE GREATER THAN 64 - One of the crop window years is < 64.

13. MONTH MUST BE BETWEEN 1 AND 12 - One of the crop window months is ≤ 0 or > 12.

14. DAY MUST BE BETWEEN 1 AND 31. - One of the crop window days is < 1 or > 31.

15. OVERLAPPING CROP WINDOWS - The start date for one of the four crop windows is > the stop date or the stop date of one crop window is ≥ the start date of the next window.

16. GROUP NUMBER MUST BE EQUAL TO 1, 2 OR 3. IGRP must have one of the three values specified above.

17. DELTA ERRORS IN BIO WINDOW PREDICTION DATES ARE NOT BETWEEN -1 AND 1. DELTWE and DELWE1 must be between -1 and +1.

18. SEGMENT SPECIFIED AS ORDINARY MUST HAVE A TRAINING SEGMENT PRIORITY LIST - Each ordinary segment must have at least 1 priority training segment specified.

19. NEITHER NAGR OR NAL CAN BE ZERO.

<u>Definitions of Local Variables:</u> I - Index in a DO loop.

Subroutine CALGEN

Purpose:

To read, check and merge the crop calendar and crop error data card files into one intermediate file for further processing. On Option A listing of the input card files is given.

Input:

ISCW, ICWE, ICALIN, ILIST, IOUT, ENDATA, ICTRY, IREG, IZONE, ISTRA, ISUB, IYR, IMO, IDAY, ITYPE, ICTRY2, IZONE 2, ISTRA2, ISUB2, DELTWE, IPLNTE, ISEGSD, ITYPE2, FATAL

Output:

NCRCD, NCRECD, NCROP, IERR, IREGL, IZONEL, ISTRL, IPRNT, GOOD1, GOOD2

Linkage:

Call CALGEN

Subroutines Used:

None

Local Variable Description

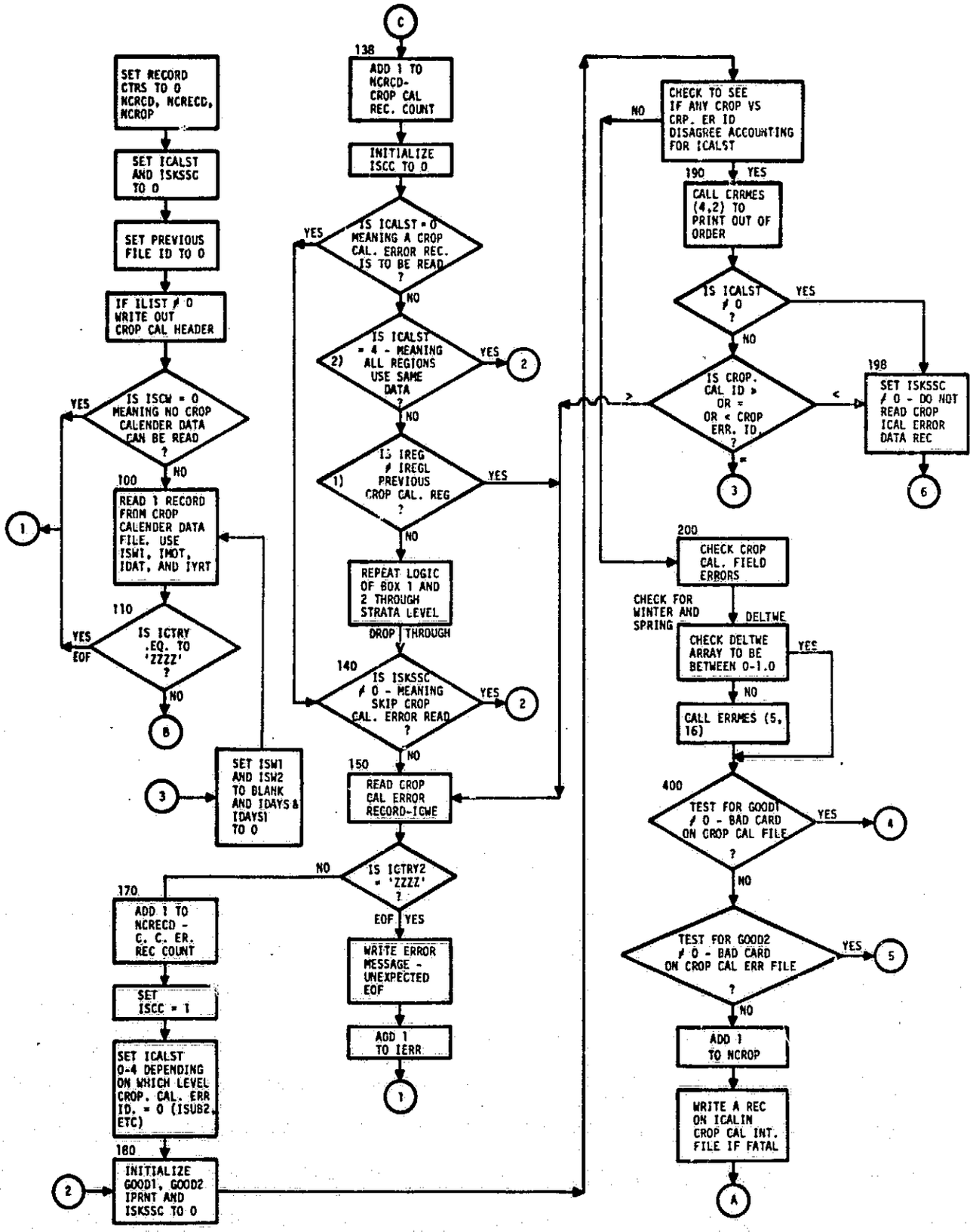
ICALST	=	0 - If for each crop calendar data card, a crop error card is to be read
	=	1 - If current crop error card is to be used for all substrata in a strata
	=	2 - All strata within a zone
	=	3 - All zone within a region
	=	4 - All region within a country
ISKSSC	-	Flag controlling reading of the crop calendar error data
	=	1 - Skip reading the crop calendar error data until the crop calendar data catches up (i.e., until the region, zone, strata, and substrata agree unless the region, zone, strata or substrata from the crop error are 0
ISCC	-	Flag to denote whether crop calendar error file was read after crop calendar data ≠0 - yes, =0 - no

ICTEM
IRTEM
IZTEM
ISTEM
ISSTEM
IMOT(8), IDAT(8), IYRT(8)

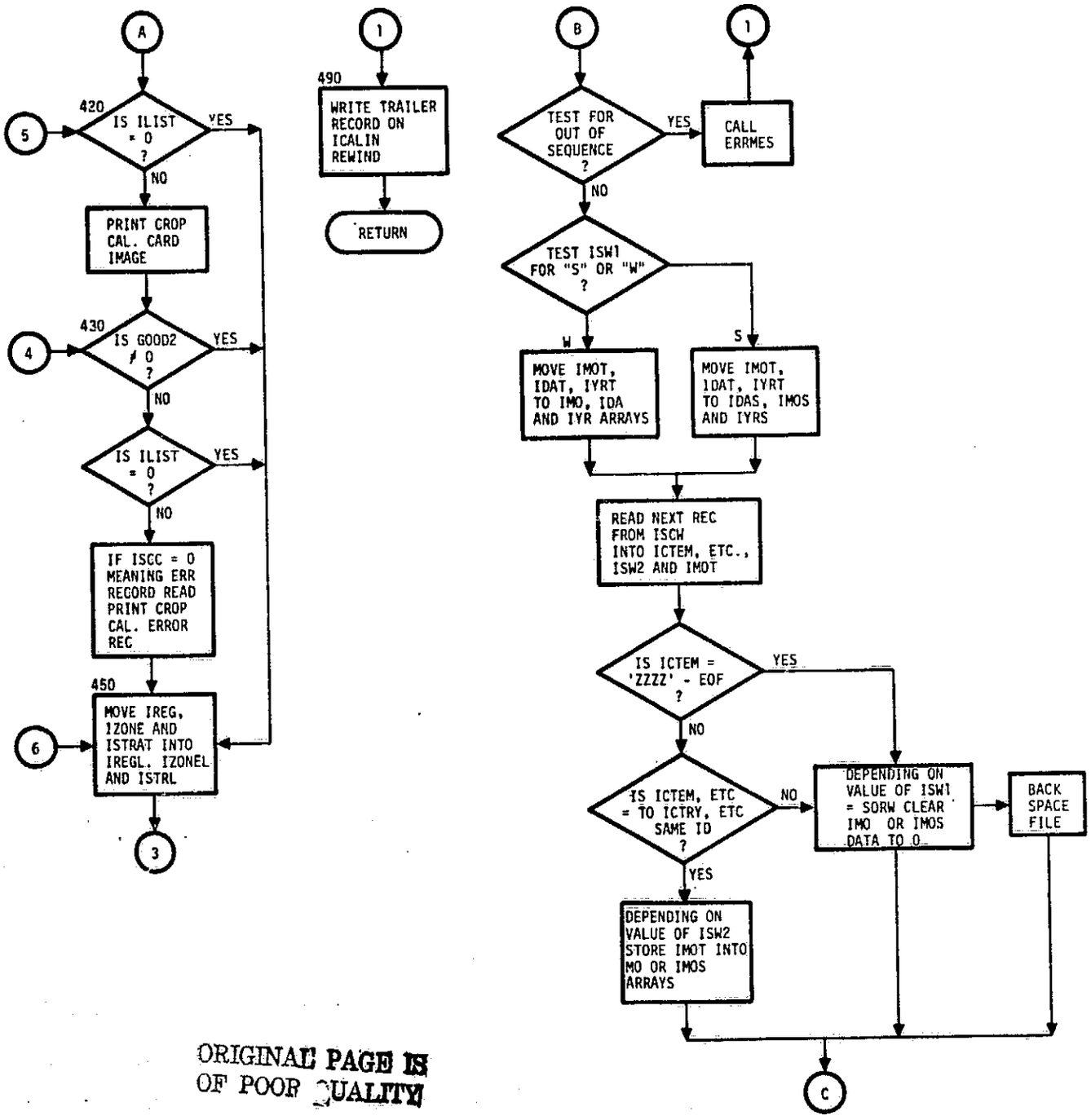
Processing:

See flowchart.

CALGEN FLOWCHART



CALGEN FLOWCHART (Continued)



SUBROUTINE CALERR

Purpose: Given the reference crop calendar, error terms and standard deviation of calendar error, this routine computes the predicted bio-window start and stop dates and their associated errors.

Input: None through COMMON
INPUT COMMON
IDAYS, IDAYS1, DELTWE, DELWE1, IPLNTE, IPLNT1,
IDAYS - $(t_o, t_f)_i$ pairs $1 \le i \le 4$

Output: CALEND COMMON
IPREDW, IPREDS, ICRPEW, ICRPES

Linkage: Call CALERR (IDAYS, DELTWE, IPLNTE, IPREDW, ICRPEW)

Subroutines Used: None

Local Variables:

$$DELIN1 = \delta_1 (t_{f_1} - t_{o_1}) + .5$$

$$DELIN2 = \delta_2 (t_{f_2} - t_{o_2}) + .5$$

$$DELIN2 = \delta_3 (t_{f_3} - t_{o_3}) + .5$$

Processing:

1. Process data

   a. If IDAYS(1) is 0, then there is no crop calendar data. Skip steps b-d.

   b. Compute DELIN1, DELIN2, DELIN3

   c. Compute predicted dates $(t_o, t_s)_i$ $1 \le i \le 4$

```
IPREDW(1) = IDAYS(1) + IPLNTE
IPREDW(2) = IDAYS(2) + DELIN1 + IPLNTE
IPREDW(3) = IDAYS(3) + DELIN1 + IPLNTE
IPREDW(4) = IDAYS(4) + DELIN1 + DELIN2 + IPLNTE
IPREDW(5) = IDAYS(5) + DELIN1 + DELIN2 + IPLNTE
IPREDW(6) = IDAYS(6) + DELIN1 + DELIN2 + DELIN3 + IPLNTE
IPREDW(7) = IDAYS(7) + DELIN1 + DELIN2 + DELIN3 + IPLNTE
IPREDW(8) = IDAYS(8) + DELIN1 + DELIN2 + DELIN3
            + DELTWE(4) * (IDAYS(8) - IDAYS(7)) + IPLNTE + .5
```

d. Compute error terms

```
ICRPEW(1) = IPLNTE
ICRPEW(2) = IPREDW(2) - IDAYS(2)
ICRPEW(3) = IPREDW(4) - IDAYS(4)
ICRPEW(4) = IPREDW(6) - IDAYS(6)
ICRPEW(5) = IPREDW(8) - IDAYS(8)
```

SUBROUTINE BETAD

Purpose: To compute a random number based on the Beta distribution or normal distribution, given a random number seed.

Input: No input from COMMON or files.

Output: No output to COMMON or files.

Linkage: CALL BETAD (SEED, XBAR, SIGMA, XI, IOPT, IER)

Input:

SEED	A double precision random number seed used to get a uniform random number P, $0 < P < 1$
XBAR	Mean value $\overline{X}$, $0 \le \overline{X} \le 1$
SIGMA	Standard deviation σ, $0 \le \sigma$
IOPT	$= 0$ use Beta distribution $\neq 0$ use normal distribution

Output:

XI	Random number based on Beta or normal distribution X_i
IER	Error flag $= 0$ no errors $= 1$ XBAR not in range, $0 \le \overline{X} \le 1$ so was reset within subroutine $= 2$ SIGMA not in range, $0 \le \sigma \le \overline{X} \sqrt{\frac{1 - \overline{X}}{\overline{X} + \epsilon}}$ so was reset within subroutine, $\epsilon = 10^{-4}$ $= 3$ Fatal error, XI could not be found within constraints of subroutine; e.g., within 35 iterations via the inverse incomplete beta function method
SEED	To be used for next call to BETAD (a double precision number)

Subroutines used:

CALL RDM1A (SEED, P) to get uniform random number P
SEED = double precision

CALL IBETAI (X, A, B, P, IER) to get incomplete beta function

Note: IBETAI is algorithm AS 63 Appl. Statist. (1973), Vol. 22, No. 3

SQRT (X)	squareroot
ALOG (X)	exponential
EXP (X)	natural logarithm
ALOG(X)	natural logarithm

Local variables:

A		First Beta parameter
B		Second Beta parameter
BP		Recalculated second Beta parameter
CHK		Normal distribution parameter
DIFF		Accuracy check
DIFF1		Check if XI close to 0
EP		10^{-4}, accuracy of answer
FLAG	INTEGER	Flag to signal XBAR > .5
H		Beta approx. parameter
I		Loop counter
K	REAL	2., method threshold constant
P		Output from RDM1A, f(X) for Beta function
PHI		Limit for iteration of P
PLO		Limit for iteration of P
PO		Output from IBETAI
R		88., Gamma constraint
RN		Normal distribution parameter
SIG		Stores SIGMA, or SIGMAL, for use in routine
SG		10^{-10} check on successive answers in loop
SIGMAL		Upper limit on SIGMA
SIGSQ		SIG * SIG, intermediate calculation
SIGT		Method threshold sigma
T		Normal distribution parameter
W		Beta approx. parameter
XAVG		XBAR, or 1 - XBAR if XBAR > .5
XHI		Limit for iteration of X
XLO		Limit for iteration of X
XSQ		XBAR * XBAR, intermediate calculation
Y		Beta approx. parameter
YP		Beta approx. parameter

BETAD Subroutine Equations

Equation set 1 - normal distribution parameters:

$$T = \begin{cases} \sqrt{\ln \frac{1}{P^2}} & 0 < P \leq .5 \\ \sqrt{\ln \frac{1}{(1-P)^2}} & .5 < P < 1 \end{cases}$$

$$CHK = T - \frac{2.30753 + .27061T}{1. + .99229T + .04481T^2}$$

$$RN = \begin{cases} -CHK & 0 < P \leq .5 \\ +CHK & .5 < P < 1 \end{cases}$$

Equation set 2 - SIGMA upper limit:

$$SIGMAL = XBAR \sqrt{\frac{1 - XBAR}{XBAR + EP}} \qquad EP = 10^{-4}$$

Equation set 3 - Beta function parameters:

$$A = \frac{XBAR^2 - XBAR\,(XBAR^2 + SIGMA^2)}{SIGMA^2}$$

$$B = \left(\frac{1 - XBAR}{XBAR}\right) A$$

Equation set 4 - method threshold:

$$SIGT = XBAR \sqrt{\frac{1 - XBAR}{XBAR + K}} \qquad K = 2$$

Equation set 5 - Beta approximation parameters:

$$YP = -RN$$

$$H = 2\left(\frac{1}{2A - 1} + \frac{1}{2B - 1}\right)^{-1}$$

$$Y = \frac{YP^2 - 3}{6}$$

$$W = \frac{YP\,(H + Y)^{1/2}}{H} - \left(\frac{1}{2B - 1} - \frac{1}{2A - 1}\right)\left(Y + \frac{5}{6} - \frac{2}{3H}\right)$$

Equation set 6 - XI for Beta approximation:

$$XI = \frac{A}{A + B \cdot e^{2W}} \quad ABS\,(ALOG\,(B) + 2 * W) \leq 87$$

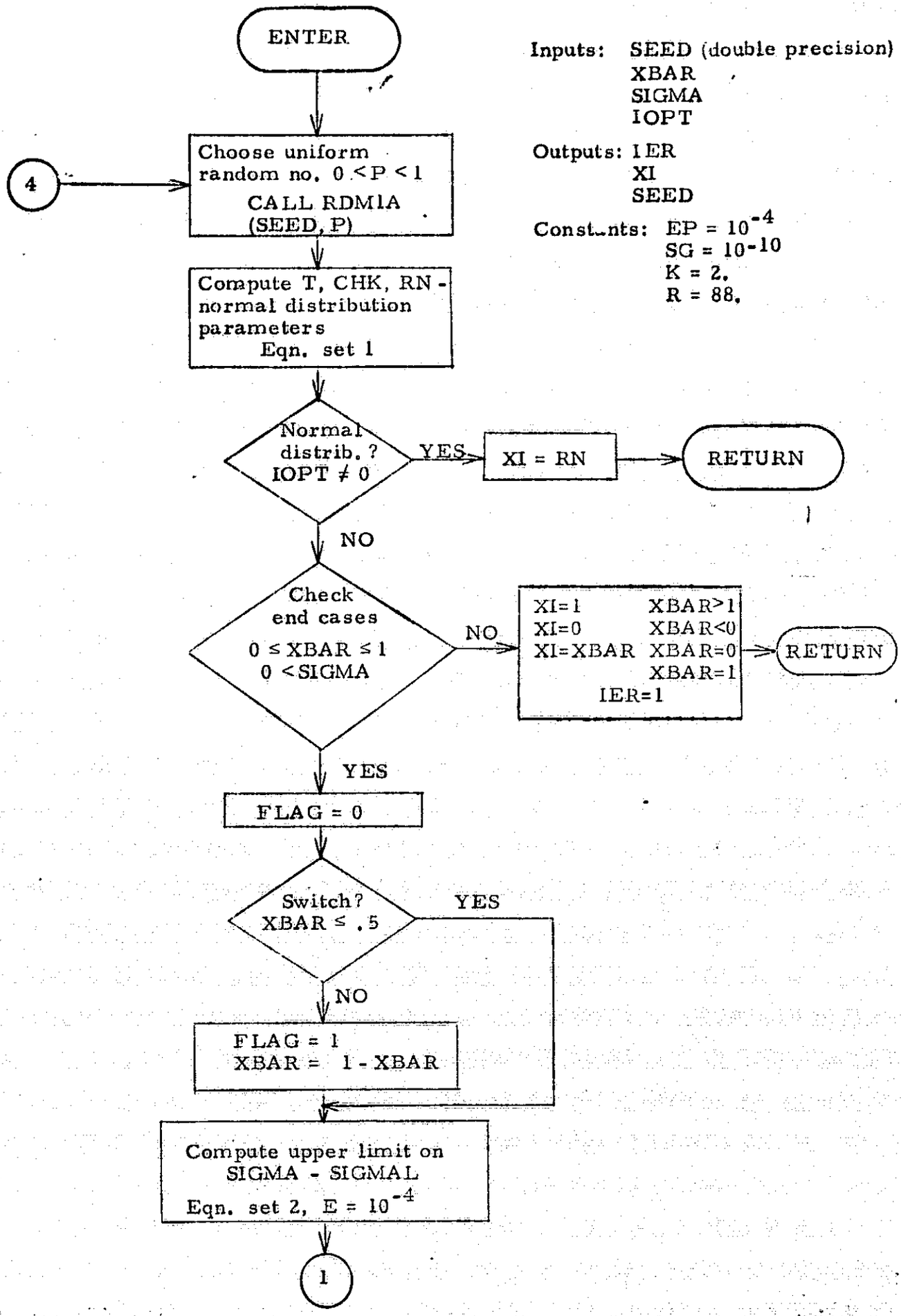
Equation set 7 - recompute A and B, Beta parameters:

$$B^1 = \frac{B}{A + B} \cdot (R - 1)$$

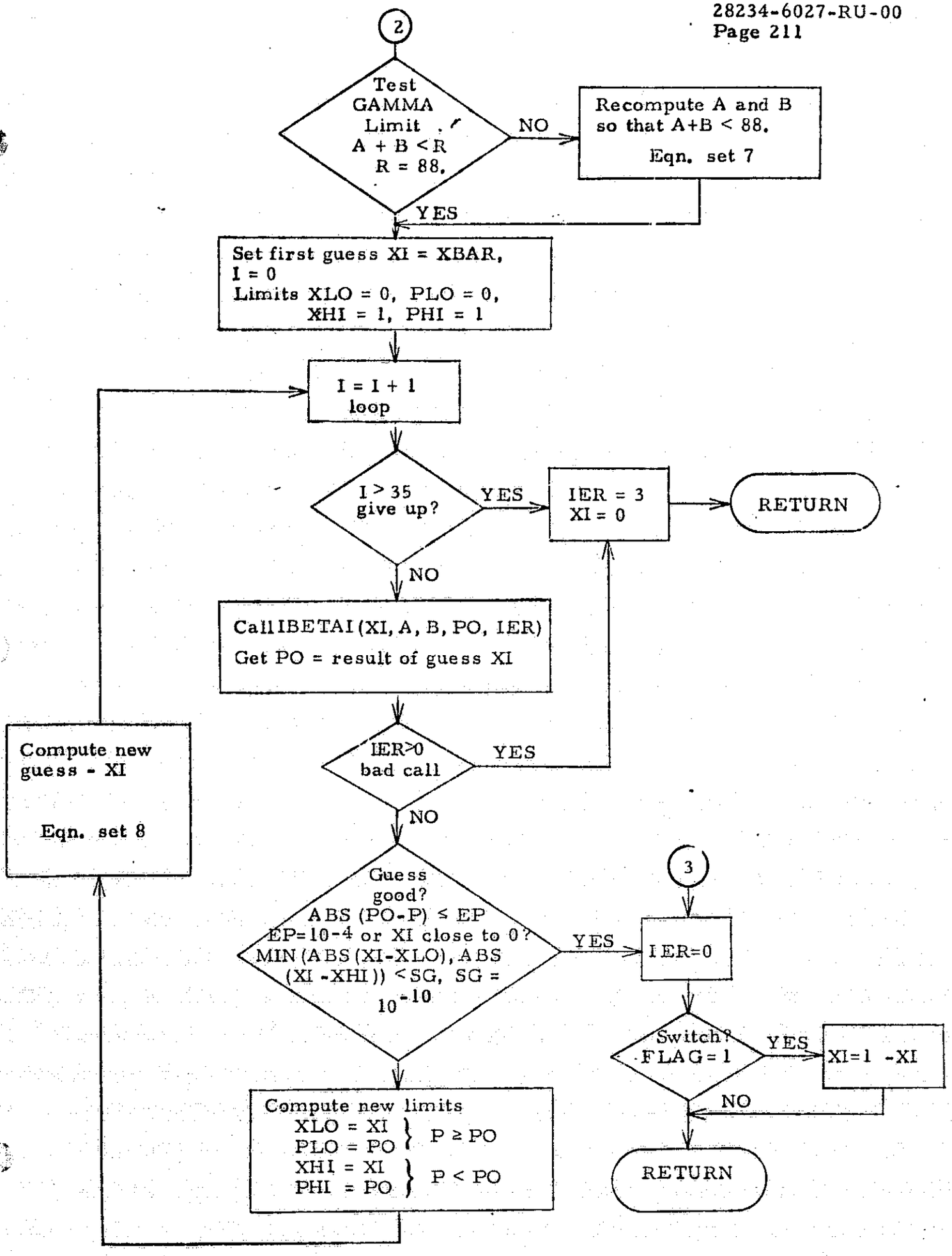
$$A^1 = \frac{BP}{B} \cdot A$$

Equation set 8 -

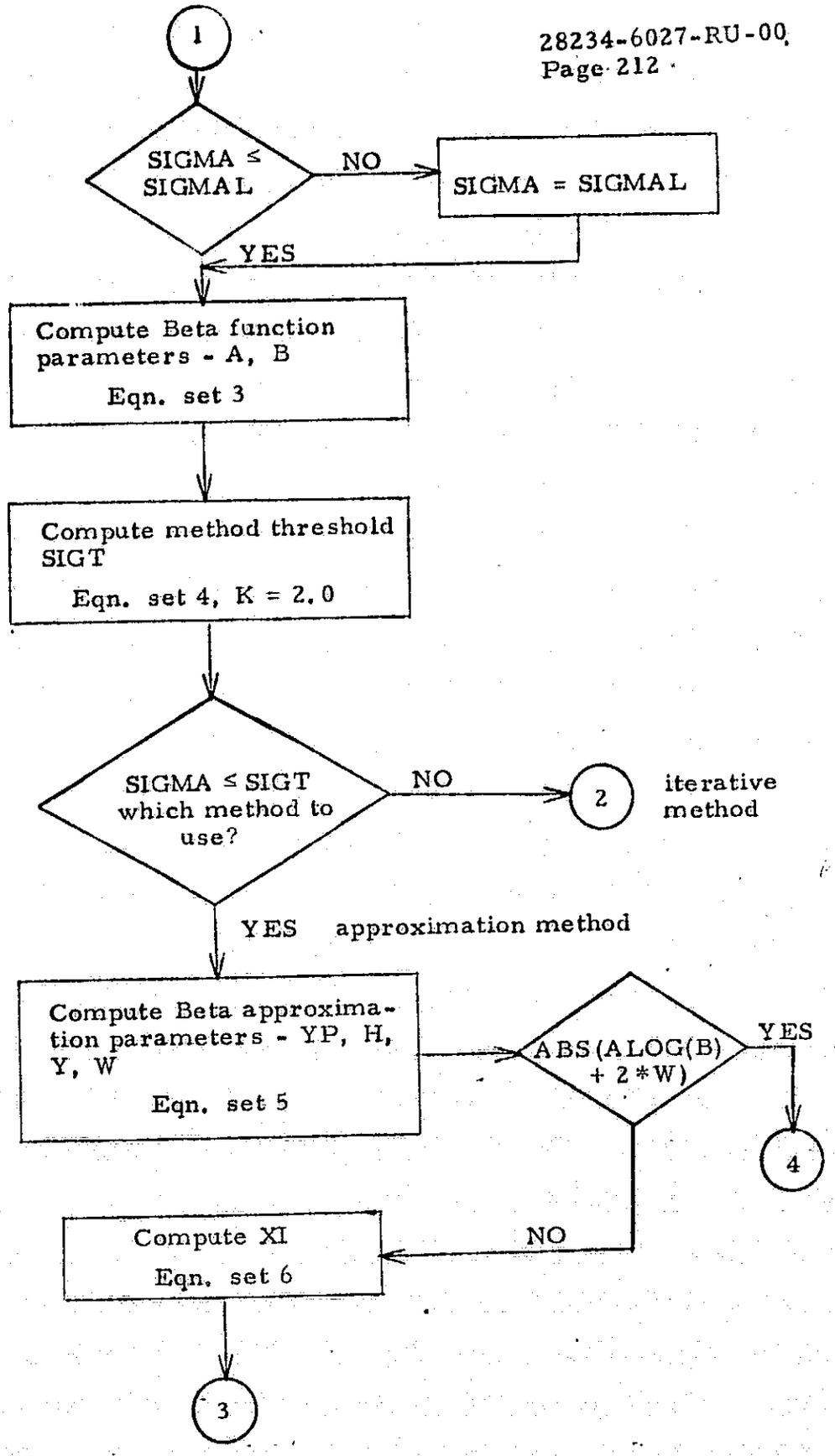
$$XI = \frac{(XH1 + XL0)}{2.}$$



BETAD Subroutine Flowchart (Sheet 1 of 3)



BETAD Subroutine Flowchart (Sheet 3 of 3)



BETAD Subroutine Flowchart (Sheet 2 of 3)

SUBROUTINE GRID

Purpose: To control the determination of the index number for a given segment latitude and longitude on the global index matrix.

Input: QLAT and OLONG in INPUT COMMON
IREG, IZONE, ISTRAT, ISUB and ISEG
IERT in calling sequence

Output: Index in input COMMON

Linkage: Call GRID (IERT)

IERT - Output error flag
=0 - o.k.
≠0 - no index could be found

Subroutines Used:

Call GRDCAL (IGRID, JGRID, INDEX, FI, FJ, IERT)
IGRID
JGRID Global grid coordinates
INDEX - Index no.
FI, FJ - Flt. pt. version of IGRID, JGRID
IERT - =0 - o.k., ≠0 - no index found

Processing:

1. Set IERT = 0.

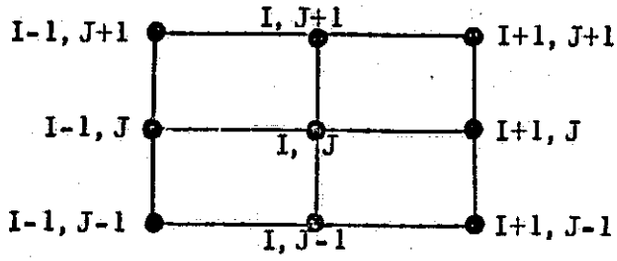
2. Compute IGRID, JGRID, FI and FJ from Equation set 1-6 in Problem Description.

3. Call GRDCAL to get index.

4. If IERT = 0, return.
   If IERT ≠ 0, print error message as defined in Problem Description and return.

SUBROUTINE GRDCAL

Purpose: To determine the most appropriate index value associated with the global grid coordinates IGRID and JGRID. The IGRID and JGRID may not have an index value assigned to it on the index matrix file. If this is the case a boundary grid is drawn about the IGRID and JGRID position and each of the eight grid points is tested in a priority order for an index value. If no index can be found, the subroutine returns control. See Gene Rasmussen's write-up "Weather Index Generation" for details of theory and method. The boundary grid looks like:



where I ≡ IGRID
J ≡ JGRID

Input: Index from file IGRDTP via COMMON DIRAC
IGRDTP in FLAG COMMON
IGRID, JGRID in CALL SEQ.

Output: Index in CALL SEQ.

Linkage: CALL GRDCAL (IGRID, JGRID, INDEX, FI, FD, IER)
IGRID, JGRID - input global grid coordinates
INDEX - Index no. (output)
IER - =0 - index found
≠0 - no index found
FI, FJ - floating point equivalents of IGRID, JGRID

Subroutines Used:

CALL RDGRID (JCOOR) - JCOOR is column coordinate ≡ rec. no.

Local Variables:

ISTATE(8, 8)	State matrix; given the state no. (the column no.), each row gives the sequence to try each coordinate of the boundary grid about IGRID, JGRID. This is a constant matrix containing bourdary matrix positions.
IBOUND(2, 8)	Boundary coordinate list. Each column of the matrix represents a grid position on the boundary grid matrix. Each row represents the X, Y coordinate and each entry is the amount to subtract or add to IGRID, JGRID to get current grid position. Used in conjunction with ISTATE.
ISTAT	State no., SLOPE - SLOPE
IPT(8)	Current states grid position sequence list

Processing:

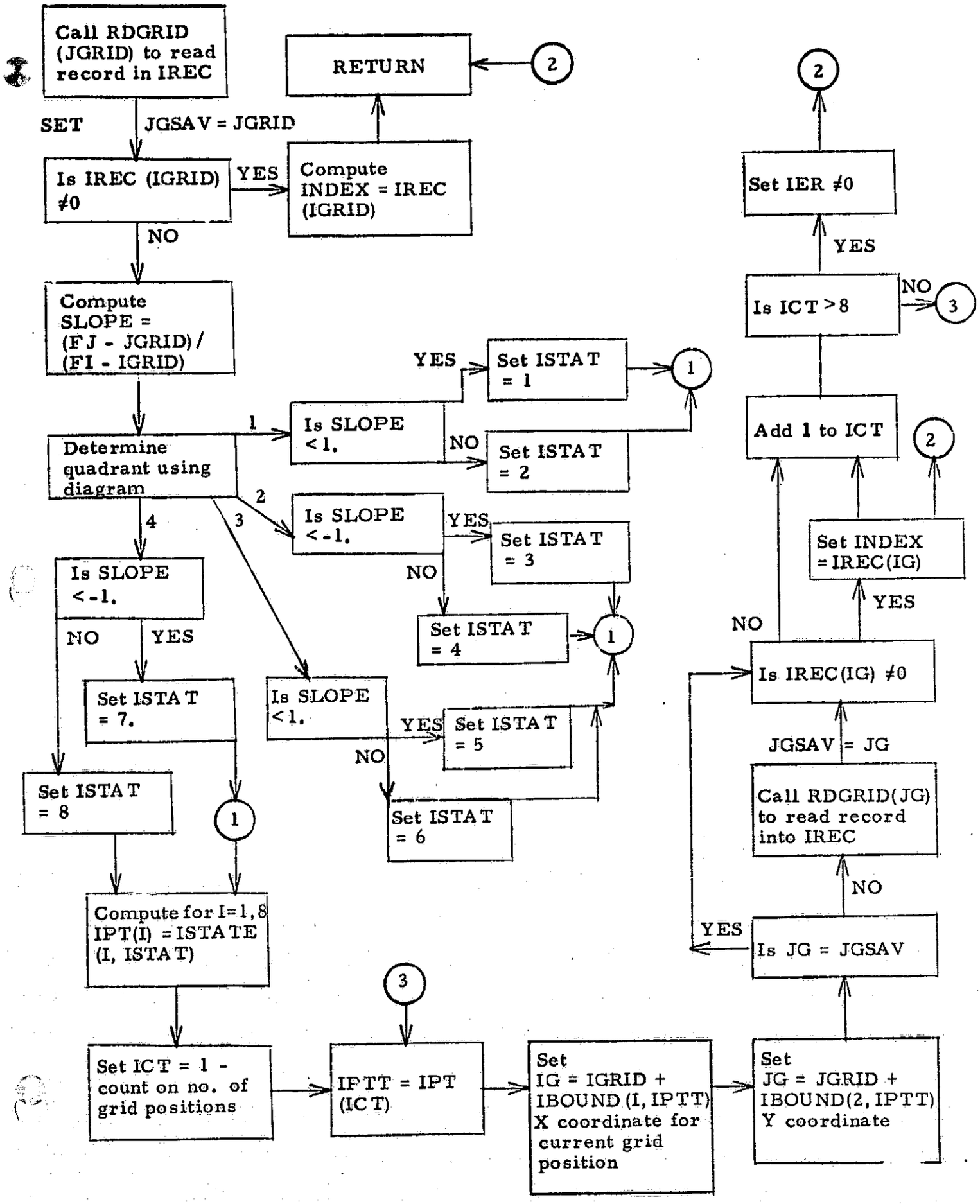
Rules for quadrant determination in boundary matrix.

(FI - IGRID)	+	-	-	+
(FJ - JGRID)	+	+	-	-
Quadrant	1	2	3	4

Rules for determining state given quadrant and SLOPE = $\frac{FJ - JGRID}{FI - IGRID}$

Quadrant	1	1	2	2	3	3	4	4
SLOPE	<1	≥1	<-1	≥-1	<1	≥1	<-1	≥-1
State	1	2	3	4	5	6	7	8

See flowchart.


Call RDGRID (JGRID) to read record in IREC
SET
JGSAV = JGRID
Is IREC (IGRID) ≠0
YES
Compute INDEX = IREC (IGRID)
RETURN
2
NO
Compute SLOPE = (FJ - JGRID) / (FI - IGRID)
Determine quadrant using diagram
1
Is SLOPE < 1.
YES
Set ISTAT = 1
1
NO
Set ISTAT = 2
2
Is SLOPE < -1.
YES
Set ISTAT = 3
NO
Set ISTAT = 4
1
3
Is SLOPE < 1.
YES
Set ISTAT = 5
NO
Set ISTAT = 6
4
Is SLOPE < -1.
NO
YES
Set ISTAT = 7.
Set ISTAT = 8
1
Compute for I=1,8 IPT(I) = ISTATE (I, ISTAT)
Set ICT = 1 - count on no. of grid positions
3
IPTT = IPT (ICT)
Set IG = IGRID + IBOUND (I, IPTT) X coordinate for current grid position
Set JG = JGRID + IBOUND(2, IPTT) Y coordinate
Is JG = JGSAV
YES
NO
Call RDGRID(JG) to read record into IREC
JGSAV = JG
Is IREC(IG) ≠0
NO
YES
Set INDEX = IREC(IG)
2
Add 1 to ICT
Is ICT > 8
NO
3
YES
Set IER ≠0
2

SUBROUTINE RDGRID

Purpose: To initialize, read a record or close the index matrix file on unit IGRDTP

Input: IGRDTP in FLAG COMMON
JG in calling seq.
ICDIN in DIRAC COMMON

Output: One record in IREC in DIRAC COMMON

Linkage: CALL RDGRID(JG)
JG is record no. or flag
If JG = 0, open the file via OPENMS, call for CDC and define file for UNIVAC
If JG = -n, close the file via CALL CLOSEMS for CDC and close for UNIVAC

Processing:

For JG = +n
Read n^{th} record from IGRDTP via CALL READMS for CDC or read for UNIVAC into IREC. 500 word record.

PART V

SUBROUTINE LISTINGS

```
* FLT ALGAMA,1,760430, 54500   , 1

000001          SUBROUTINE ALGAMA(X,GAMMA,IER)                                  ALGAMA
000002    C      CALCULATES THE GAMMA FUNCTION                                  ALGAMA
000003    C                                                                     ALGAMA
000004    C   X      INPUT                                                      ALGAMA
000005    C   GAMMA  OUTPUT                                                     ALGAMA
000006    C   IER    ERROR FLAG = 60 IF X NOT IN RANGE 0. TO 88.                ALGAMA
000007    C                                                                     ALGAMA
000008          DIMENSION B(14),C(6)                                            ALGAMA
000009          DOUBLE PRECISION  B,C,F,G,HLNPI,Y,Z
000010          DATA B/-3.03019081028D-6,2.79832889938D-5,-1.21417348706 32D-4,  ALGAMA  *NEW
000011         *3.75366052263070-4,-8.37564685135170-4,2.06109185022754D-3,     ALGAMA  *NEW
000012         *-2.83646252037282D-3,1.11497143357789D-2,-2.66186594953 0D-4,   ALGAMA  *NEW
000013         *7.42489154194447D-2,8.15769261241555D-2,4.11840330168781D-1,    ALGAMA  *NEW
000014         *4.22784335102335D-1,9.99999999999990D-1/                        *NEW
000015          DATA C/-1.91752691752692D-3,8.41750841750842D-4,-5.95238095238095D ALGAMA  *NEW
000016         *-4,7.93650793650794D-4,-2.77777777777778D-3,8.33333333333333D-2/ ALGAMA  **-7
000017          DATA HLNPI/.91893853320467D/                                    ALGAMA
000018          GAMMA=0.                                                        ALGAMA
000019          IER=60                                                          ALGAMA
000020          IF (X .LE. 0.0 .OR. X .GT. 88.0) RETURN                         ALGAMA
000021          IER=0                                                           ALGAMA
000022          Y=X                                                             ALGAMA
000023          IF(Y.GE.10.) GO TO 60                                           ALGAMA
000024          Z=DMOD(Y,1.0)                                                   ALGAMA
000025          IF(Z.NE.0.0) GO TO 20                                           ALGAMA
000026          Z=1.0                                                           ALGAMA
000027          G=2.0                                                           ALGAMA
000028          GO TO 27                                                        ALGAMA
000029    20    G=B(1)                                                          ALGAMA
000030          DO 26 J=2,14                                                    ALGAMA
000031    26    G=G*Z+B(J)                                                      ALGAMA
000032    27    IF(Y-3.0) 28,45,38                                              ALGAMA
000033    28    IF(Y.GT.2.0) GO TO 45                                           ALGAMA
000034          G=G/Y                                                           ALGAMA
000035          IF(Y.LE.1.0) G=G/(Y+1.0)                                        ALGAMA
000036          GO TO 45                                                        ALGAMA
000037    38    F=1.0                                                           ALGAMA
000038          K = DINT(Y-7+.5)-2                                              *NEW
000039          DO 40 J=1,K                                                     ALGAMA  **-1
000040          Y=Y-1.0                                                         ALGAMA
000041    40    F=F*Y                                                           ALGAMA
000042          G=G*F                                                           ALGAMA
000043    45    GAMMA=DLOG(G)                                                   ALGAMA
000044          RETURN                                                          ALGAMA
000045    60    Z=Y*Y                                                           ALGAMA
000046          G=C(1)                                                          ALGAMA
000047          DO 65 J=2,6                                                     ALGAMA
000048    65    G=G/Z+C(J)                                                      ALGAMA
000049          G=((G/Y+HLNPI)+((Y-.5)*DLOG(Y)-Y))                              ALGAMA
000050          GAMMA=G                                                         ALGAMA
000051          RETURN                                                          ALGAMA
000052          END                                                             ALGAMA
```

```
* ELT BETAD,1,760401, 67415     , 1

000001          SUBROUTINE BETAD(SEED,XBAR,SIGMA,XI,IOPT,IER)                BETAD
000002          DOUBLE PRECISION SEED                                        BETAD
000003          INTEGER FLAG                                                 BETAD
000004          REAL K                                                       BETAD
000005          DATA R/88./                                                  BETAD
000006          DATA K/2./                                                   BETAD
000007          DATA FP/.00005/                                              BETAD
000008          DATA SG /1.E-8 /                                             BETAD
000009          DATA FLAG/0/                                                 BETAD
000010          DATA I/0/                                                    BETAD
000011       50 CONTINUE                                                     MODS
000012          FLAG =0                                                      MODS
000013          I = 0                                                        MODS
000014          XAVG=XBAR                                                    BETAD
000015          XI=0.0                                                       BETAD
000016          IER=0                                                        BETAD
000017     C                                                                 BETAD
000018     C  CHOOSE UNIFORM RANDOM NUMBER                                   BETAD
000019          CALL RDM1A(SEED,P)                                           BETAD
000020     C                                                                 BETAD
000021     C COMPUTE T,CHK,RN, - NORMAL DISTRIB. PARAMETERS                  BETAD
000022          T=SQRT(ALOG(1.0/(P*P)))                                      BETAD
000023          IF(P.GT.0.5)T=SQRT(ALOG(1.0/((1.0-P)*(1.0-P))))              BETAD
000024          CHK=T-(2.30753+.27061*T)/(1.0+0.99229*T+.04481*T*T)          BETAD
000025          RN=CHK                                                       BETAD
000026          IF(P.LE.0.5) RN=-RN                                          BETAD
000027     C                                                                 BETAD
000028     C  IF NORMAL DISTRIB. OPTION, JUMP OUT                            BETAD
000029          IF(IOPT.EQ.0) GO TO 70                                       BETAD
000030          XI=RN                                                        BETAD
000031          RETURN                                                       BETAD
000032       70 CONTINUE                                                     BETAD
000033     C                                                                 BETAD
000034     C  CHECK FOR END CASES                                            BETAD
000035          IF(XBAR.LT.0. .OR.XBAR.GT.1.) IER=1                          BETAD
000036          IF(SIGMA.LT.0.0) IER=2                                       BETAD
000037          IF (XBAR.GT.0.0.AND.XBAR.LT.1.0.AND.SIGMA.GT.0.0) GO TO 10   BETAD
000038          XI=XBAR                                                      BETAD
000039          IF(XBAR .LT. 0.)XI = 0.0                                     MODS
000040          IF(XBAR .GT. 1.)XI = 1.0                                     MODS
000041          RETURN                                                       BETAD
000042     C                                                                 BETAD
000043     C SWITCH IF AVERAGE ABOVE HALF                                    BETAD
000044       10 CONTINUE                                                     BETAD
000045          IF (XBAR.LE.0.5) GO TO 20                                    BETAD
000046          FLAG=1                                                       BETAD
000047          XAVG=1.0-XBAR                                                BETAD
000048     C                                                                 BETAD
000049     C  COMPUTE UPPER LIMIT ON SIGMA                                   BETAD
000050       20 CONTINUE                                                     BETAD
000051          SIGMAL=XAVG*SQRT((1.00-XAVG)/(XAVG+FP))                      BETAD
000052          SIG=SIGMA                                                    BETAD
000053          IF(SIGMA.LE.SIGMAL) GO TO 30                                 BETAD
000054          SIG=SIGMAL                                                   BETAD
000055          IER=2                                                        BETAD
000056     C                                                                 BETAD
000057     C  COMPUTE THE METHOD THRESHHOLD                                  BETAD
000058       30 CONTINUE                                                     BETAD
```

```
000059          SIGT=XAVG*SQRT((1.0-XAVG)/(XAVG*K))                     BETAD
000060    C                                                             BETAD
000061    C   COMPUTE BETA PARAMETERS A AND B                           BETAD
000062          XSQ=XAVG*XAVG                                           BETAD
000063          SIGSQ=SIG*SIG                                           BETAD
000064          A=(XSQ-XAVG*(XSQ+SIGSQ))/SIGSQ                          BETAD
000065          B=((1.0-XAVG)/XAVG)*A                                   BETAD
000066          IF(SIG.GT.SIGT) GO TO 40                                BETAD
000067    C                                                             BETAD
000068    C   APPROXIMATION METHOD                                      BETAD
000069    C                                                             BETAD
000070    C                                                             BETAD
000071    C   COMPUTE BETA APPROXIMATION PARAMETERS                     BETAD
000072          YP=-RN                                                  BETAD
000073          H=2.0/ (1.0/(2.0*A- 1.0) + 1.0/(2.0*B - 1.0))           BETAD
000074          Y=(YP*YP-3.0)/6.0                                       BETAD
000075          H=(YP*SQRT(H+Y))/                                       BETAD
000076        1  H - (1.0/(2.0*B-1.0)-1.0/(2.0*A-1.0))*(Y+5.0/6.0-2.0/(3.0*H))  BETAD
000077    C                                                             BETAD
000078    C   COMPUTE XI                                                BETAD
000079          IF(ABS(ALOG(B) + 2.*H) .GT. 87.0)GO TO 50               NOOS
000080          XI=A/(A+B*EXP(2.0*H))                                   BETAD
000081          GO TO 60                                                BETAD
000082    C                                                             BETAD
000083    C   ITERATIVE METHOD                                          BETAD
000084    C                                                             BETAD
000085    C RECOMPUTE A AND B IF OVER LIMIT                             BETAD
000086       40 CONTINUE                                                BETAD
000087          CHK=A+B                                                 BETAD
000088          IF(CHK.LT.R) GO TO 80                                   BETAD
000089          BP=(B/CHK)*(R-1.)                                       BETAD
000090          A=(BP/B)*A                                              BETAD
000091          B=BP                                                    BETAD
000092       80 CONTINUE                                                BETAD
000093          I=0                                                     BETAD
000094          PHI=1.                                                  BETAD
000095          PLO=0.                                                  BETAD
000096          XHI=1.                                                  BETAD
000097          XLO=0.                                                  BETAD
000098          XI=XAVG                                                 BETAD
000099      120 I=I+1                                                   BETAD
000100          IF(I.GT.35) GO TO 100                                   BETAD
000101          CALL TBETAI(XI,A,B,PO,IER)                              BETAD
000102          IF(IER.GT.0) GO TO 100                                  BETAD
000103          DIFF=ABS(PO-P)                                          BETAD
000104          DIFF1=AMIN1(ABS(XI-XLO),ABS(XI-XHI))                    BETAD
000105          IF(DIFF.LE.EP.OR.DIFF1.LT.SG ) GO TO 60                 BETAD
000106    C                                                             BETAD
000107    C   CHECK IF LONG, WITHIN TOLERANCE                           BETAD
000108          IF(PO.LT.P) GO TO 105                                   BETAD
000109          XHI=XI                                                  BETAD
000110          PHI=PO                                                  BETAD
000111          GO TO 106                                               BETAD
000112      105 CONTINUE                                                BETAD
000113          XLO=XI                                                  BETAD
000114          PLO=PO                                                  BETAD
000115      106 CONTINUE                                                BETAD
000116          XI = (XHI + XLO)/2.                                     NOOO
000117          GO TO 120                                               BETAD
000118    C ERROR RETURN                                                BETAD
```

```
000119          100 CONTINUE                                          BETAD
000120              IER=5                                             BETAD
000121              XI=0.0                                            BETAD
000122              RETURN                                            BETAD
000123           60 CONTINUE                                          BETAD
000124              IF(FLAG.EQ.1) XI=1.0 - XI                         BETAD
000125              RETURN                                            BETAD
000126              END                                               BETAD
```

```
@ ELT CALERR,1,760401, 67397    , 1

      SUBROUTINE CALERR(IDAYS,DELTWE,IPLNTE,IPREDW,ICRPEW)                CALERR
C     COMPUTES PREDICTED BIO WINDOW START AND STOP DATES WITH THEIR       CALERR
C     ASSOCIATED ERRORS.                                                  CALERR
      DIMENSION IDAYS(8),DELTWE(4),IPREDW(8),ICRPEW(5)                    CALERR
      IF(IDAYS(1) .EQ. 0)GO TO 50                                         CALERR
      DELIN1 = DELTWE(1)*(IDAYS(2) - IDAYS(1)) + .5                       CALERR
      DELIN2 = DELTWE(2)*(IDAYS(4) - IDAYS(3)) + .5                       CALERR
      DELIN3 = DELTWE(3)*(IDAYS(6) - IDAYS(5)) + .5                       CALERR
      IPREDW(1) = IDAYS(1) + IPLNTE                                       CALERR
      IPREDW(2) = IDAYS(2) + DELIN1 + IPLNTE                              CALERR
      IPREDW(3) = IDAYS(3) + DELIN1 + IPLNTE                              CALERR
      IPREDW(4) = IDAYS(4) + DELIN1 + DELIN2 + IPLNTE                     CALERR
      IPREDW(5) = IDAYS(5) + DELIN1 + DELIN2 + IPLNTE                     CALERR
      IPREDW(6) = IDAYS(6) + DELIN1 + DELIN2 + DELIN3 + IPLNTE            CALERR
      IPREDW(7) = IDAYS(7) + DELIN1 + DELIN2 + DELIN3 + IPLNTE            CALERR
      IPREDW(8) = IDAYS(8) + DELIN1 + DELIN2 + DELIN3 + DELTWE(4)         CALERR
     1*(IDAYS(8) - IDAYS(7)) + .5 + IPLNTE                                MPD3
      ICRPEW(1) = IPLNTE                                                  CALERR
      ICRPEW(2) = IPREDW(2) - IDAYS(2)                                    CALERR
      ICRPEW(3) = IPREDW(4) - IDAYS(4)                                    CALERR
      ICRPEW(4) = IPREDW(6) - IDAYS(6)                                    CALERR
      ICRPEW(5) = IPREDW(8) - IDAYS(8)                                    CALERR
   50 RETURN                                                              CALERR
      END                                                                 CALERR
```

```
p FLT CALGEN,1,760401, 67395   , 1

000001          SUBROUTINE CALGEN                                       CALGEN
000002    C     THIS ROUTINE READS,CHECKS AND MERGES THE CROP CALENDER AND CROP   CALGEN
000003    C     CALENDER AND CROP ERROR DATA CARD FILES INTO AN INTERMEDIATE FILE.CALGEN
000004          COMMON /CONST/                                          CONST
000005         1PI,DEGTOR,      SLATMX,ENDATA,ZERO,INP,IOUT,INTSUB,INTSEG,ISIDF,  CONST
000006         2ICRF,ISHF,LRSIDF,LRCHF,LRSHF,IWIN,MXSEG,ICALIN,ICHE,AREALF       CONST
000007         3,ISW1,ISW2,IGRDTP                                       MODB
000008          INTEGER ZERO                                            CONST
000009          COMMON /FLAGS/                                          FLAGS
000010         1IERR,IPRNT,GOOD1,NSHC,NSSC,NSEGC,NSUBR,NSEGR,NRSIDF,NRCHF,NRSHF,  FLAGS
000011         2ISSC,ISTAT,IREGL,IZONEL,ISTRL,ISUBL,ISEGL,NSEG,IMXSEG,NTLIST,     FLAGS
000012         3READSF,GOOD2,FATAL,IDAYS(8),IDSEG(150)                  FLAGS
000013         1,IMO1(8),IDA1(8),IYR1(8),IYR1(8),IMO1(8),IDA1(8)        MOD2
000014          INTEGER GOOD1,GOOD2,READSF,FATAL                        FLAGS
000015          COMMON /INPUT/                                          INPUT
000016         1RSEED,SEBAR,HISTPW,DEVTPH,CV(3),DEVTPH,OLAT,OLONG,SLAT(3),SLONG(3)INPUT
000017         2,NAMSUB(2),ICTRY,ICTRY2,ICASE,NLIST,ITSEG,ISHD,ISSD,ISID,ISCH,    INPUT
000018         3ISEG,IREG,IZONE,ISTRAT,ISUB,ISH,ITYPE,ITYPE2,INDEX,ISEG2,IREG2,   INPUT
000019         4IZONE2,ISTRA2,ISUB2,LAT(3),LONG(3),IYR(8),IMO(8),IDA(8),CV4,IGRP, INPUT
000020         5DELTHE(4),IPLNTT,ISEGSD,ITRIND,NAGR,NAL,ITLIST(6),NCRCD,NCRECD,   INPUT
000021         6NCROP,IDAYS1(8),DELTHE1(4),IPLNT1,ISGSD1,IDAS(8),IMOS(8),IYRS(8)  INPUT
000022          DOUBLE PRECISION RSEED                                  INPUT
000023          REAL NAMSUB,ICTRY,ICTRY2                                INPUT
000024          DATA IBK/1H /                                           MOD2
000025          REAL ICTEM                                              CALGEN
000026          NCRCD = 0                                               CALGEN
000027          NCRECD = 0                                              CALGEN
000028          NCROP = 0                                               CALGEN
000029          ICALST = 0                                              CALGEN
000030          ISHSSF = 0                                              CALGEN
000031          IREGL = 0                                               CALGEN
000032          IZONEL = 0                                              CALGEN
000033          ISUBL = 0                                               MOD4
000034          ISTRL = 0                                               CALGEN
000035          IF(ILIST .NE. 0)WRITE(IOUT,10)                          CALGEN
000036       10 FORMAT(50H1  SUBSTRATA CROP CALENDER AND CALENDER ERROR DATA/)  CALGEN
000037          IF(ISCH .EQ. 0)GO TO 490                                CALGEN
000038      100 ISW1 = IBK                                              MOD2
000039          ISW2 = IBK                                              MOD2
000040          DO 105 I=1,8                                            MOD2
000041          IDAYS(I) = 0                                            CALGEN
000042          IDAYS1(I) = 0                                           CALGEN
000043      105 CONTINUE                                                CALGEN
000044          READ(ISCH,1)ICTRY,IREG,IZONE,ISTRAT,ISUB,ISH1,(IYRT(I),IMOT(I),   CALGEN
000045         1IDAT(I),I=1,8),ITYPE                                    CALGEN
000046        1 FORMAT(5X,A4,2I4,2I5,A1,3(6I2,1X),6I2,I1)               CALGEN
000047      110 IF(ICTRY .EQ. ENDATA)GO TO 490                          CALGEN
000048          IF(IREG   -  IREGL)480,111,114                          MOD4
000049      111 IF(IZONE   -  IZONEL)480,112,114                        MOD4
000050      112 IF(ISTRAT   -  ISTRL)480,113,114                        MOD4
000051      113 IF(ISUB   -  ISUBL)480,480,114                          MOD4
000052      114 CONTINUE                                                MOD4
000053          IF(ISW1 .EQ. 1HS)GO TO 120                              CALGEN
000054          DO 115 I=1,8                                            CALGEN
000055          IDA(I) = IDAT(I)                                        CALGEN
000056          IMO(I) = IMOT(I)                                        CALGEN
000057          IYR(I) = IYRT(I)                                        CALGEN
000058      115 CONTINUE                                                CALGEN
```

```
000059          GO TO 125                                                CALGEN
000060      120 DO 122 I=1,8                                             CALGEN
000061          IDAS(I) = IDAT(I)                                        CALGEN
000062          IMOS(I) = IMOT(I)                                        CALGEN
000063          IYRS(I) = IYRT(I)                                        CALGEN
000064      122 CONTINUE                                                 CALGEN
000065      125 READ(ISCW,1)ICTEM,IFTEM,IZTEM,ISTEM,ISSTEM,ISW2,(IYR1(I),  MOD2
000066         1IMO1(I),IDA1(I),I=1,8),ITYPE1                            MOD2
000067          IF(ICTEM .EQ. ENDATA)GO TO 130                           CALGEN
000068          IF(ICTEM .NE. ICIRY)GO TO 130                            CALGEN
000069          IF(IRTEM .NE. IREG) GO TO 130                            CALGEN
000070          IF(IZTEM .NE. IZONE)GO TO 130                            MOD5
000071          IF(ISTEM .NE. ISTRAT)GO TO 130                           CALGEN
000072          IF(ISSTEM .NE. ISUB)GO TO 130                            CALGEN
000073          IF(ISW2 .EQ. IHS)GO TO 128                               CALGEN
000074          DO 126 I=1,8                                             CALGEN
000075          IDA (I) = IDA1(I)                                        MOD2
000076          IMO(I) = IMO1(I)                                         MOD2
000077          IYR(I) = IYR1(I)                                         MOD2
000078      126 CONTINUE                                                 CALGEN
000079          GO TO 130                                                CALGEN
000080      128 DO 129 I=1,8                                             CALGEN
000081          IDAS(I) = IDA1(I)                                        MOD2
000082          IMOS(I) = IMO1(I)                                        MOD2
000083          IYRS(I) = IYR1(I)                                        MOD2
000084      129 CONTINUE                                                 CALGEN
000085          GO TO 130                                                CALGEN
000086      130 ISW2 = TEK                                               MOD3
000087          DO 131 I=1,8                                             MOD3
000088          IDA1(I) = 0                                              MOD3
000089          IMO1(I) = 0                                              MOD3
000090          IYR1(I) = 0                                              MOD3
000091      131 CONTINUE                                                 MOD3
000092          ITYPE1 = 0                                               MOD3
000093          IF(ISW1 .EQ. IHW)GO TO 133                               MOD3
000094          DO 132 I=1,8                                             CALGEN
000095          IDA(I) = 0                                               CALGEN
000096          IMO(I) = 0                                               CALGEN
000097          IYR(I) = 0                                               CALGEN
000098      132 CONTINUE                                                 CALGEN
000099          GO TO 135                                                CALGEN
000100      133 DO 134 I=1,8                                             CALGEN
000101          IDAS(I) = 0                                              CALGEN
000102          IMOS(I) = 0                                              CALGEN
000103          IYRS(I) = 0                                              CALGEN
000104      134 CONTINUE                                                 CALGEN
000105      135 BACKSPACE ISCW                                           CALGEN
000106      138 NCRCD = NCRCD + 1                                        CALGEN
000107          ISCC = 0                                                 CALGEN
000108          IF(ICALST .EQ. 0)GO TO 140                               CALGEN
000109          IF(ICALST .EQ. 4)GO TO 180                               CALGEN
000110          IF(IREG .NE. IREGL)GO TO 150                             CALGEN
000111          IF(ICALST .EQ. 3)GO TO 180                               CALGEN
000112          IF(IZONE .NE. IZONE1)GO TO 150                           CALGEN
000113          IF(ICALST .EQ. 2)GO TO 180                               CALGEN
000114          IF(ISTRAT .EQ. ISTR1)GO TO 180                           CALGEN
000115      140 IF(ISWSS1 .NE. 0)GO TO 180                               CALGEN
000116      150 READ(ICRY,2)ICTRY2,IREG2,IZONE2,ISTRA2,ISUB2,(DELTHF(I),I=1,4),  CALGEN
000117         1IPLNT1,ISGSD1,(PEL+F1(I),I=1,4),IPLNT1,ISGSD1,ITYPE2     CALGEN
000118         2 FORMAT(5X,A4,2I4,2I5,2(4F5.5,I3,I2),2X,I1)              MOD5
```

```
      IF(ICTRY2 .NE. FNDATA)GO TO 170                                   CALGEN
      WRITE(IOUT,21)                                                    CALGEN
   21 FORMAT(//44H UNEXPECTED EOF ON CROP CALENDER ERROR FILE.)         MOD3
      IERR = IERR + 1                                                   CALGEN
      GO TO 490                                                         CALGEN
  170 NCRECD = NCRECD + 1                                               CALGEN
      ISKC = 1                                                          CALGEN
      ICALST = 0                                                        CALGEN
      IF(ISUB2 .EQ. 0) ICALST = 1                                       CALGEN
      IF(ISTRA2 .EQ. 0)ICALST = 2                                       CALGEN
      IF(IZONE2 .EQ. 0)ICALST = 3                                       CALGEN
      IF(IREG2  .EQ. 0)ICALST = 4                                       CALGEN
  180 GOOD1 = 0                                                         CALGEN
      GOOD2 = 0                                                         CALGEN
      IPRNT = 0                                                         CALGEN
      ISKSSC = 0                                                        CALGEN
      IF(ICTRY .NE. ICTRY2)GO TO 190                                    CALGEN
      IF(ICALST .EQ. 4)GO TO 200                                        CALGEN
      IF(IREG .NE. IREG2)GO TO 190                                      CALGEN
      IF(ICALST .EQ. 3)GO TO 200                                        CALGEN
      IF(IZONE .NE. IZONE2)GO TO 190                                    CALGEN
      IF(ICALST .EQ. 2)GO TO 200                                        CALGEN
      IF(ISTRA1 .NE. ISTRA2)GO TO 190                                   CALGEN
      IF(ICALST .EQ. 1)GO TO 200                                        CALGEN
      IF(ISUB .EQ. ISUB2)GO TO 200                                      CALGEN
  190 CALL ERRMES(4,2)                                                  CALGEN
      IF(ICALST .EQ. 4)GO TO 198                                        CALGEN
      IF(IREG - IREG2)198,192,150                                       CALGEN
  192 IF(ICALST .EQ. 3)GO TO 198                                        CALGEN
      IF(IZONE - IZONE2)198,194,150                                     CALGEN
  194 IF(ICALST .EQ. 2)GO TO 198                                        CALGEN
      IF(ISTRA1 - ISTRA2)198,196,150                                    CALGEN
  196 IF(ISUB - ISUB2)198,100,150                                       CALGEN
  198 ISKSSC = 1                                                        CALGEN
      GO TO 450                                                         CALGEN
  200 IPRNT = 0                                                         CALGEN
  205 IF(IYR(1) .EQ. 0)GO TO 225                                        CALGEN
      DO 250 I=1,8                                                      CALGEN
      IF(IYR(I) .LT. 64)CALL ERRMES(4,12)                               CALGEN
      IF(IMO(I) .LT. 1 .OR. IMO(I) .GT. 12)CALL ERRMES(4,13)            CALGEN
      IF(IDA(I) .LT. 1 .OR. IDA(I) .GT. 31)CALL ERRMES(4,14)            CALGEN
      FIDA = IDA(I)                                                     CALGEN
      CALL LIPA(FIDA,IMO(I),IYR(I),DAYS)                                CALGEN
      IDAYS(I) = DAYS                                                   CALGEN
      IF(I  .EQ. 1)GO TO 250                                            CALGEN
      IF(MOD(I,2) .EQ. 1)GO TO 240                                      CALGEN
      IF(IDAYS(I-1) .GT. IDAYS(I))CALL ERRMES(4,15)                     CALGEN
      GO TO 250                                                         CALGEN
  240 IF(IDAYS(I-1) .GE. IDAYS(I))CALL ERRMES(4,15)                     CALGEN
  250 CONTINUE                                                          CALGEN
  225 IF(IYRS(1) .EQ. 0)GO TO 255                                       CALGEN
      DO 235 I=1,8                                                      CALGEN
      IF(IYRS(I).LT. 64)CALL ERRMES(4,12)                               CALGEN
      IF(IMOS(I) .LT. 1 .OR. IMOS(I) .GT. 12)CALL ERRMES(4,13)          CALGEN
      IF(IDAS(I) .LT. 1 .OR. IDAS(I) .GT. 31)CALL ERRMES(4,14)          CALGEN
      FIDA1 = IDAS(I)                                                   CALGEN
      CALL LIPA(FIDA1,IMOS(I),IYRS(I),DAYS1)                            CALGEN
      IDAYS1(I) = DAYS1                                                 CALGEN
      IF(I  .EQ. 1)GO TO 235                                            CALGEN
      IF(MOD(I,2) .EQ. 1)GO TO 230                                      CALGEN
```

```
000179          IF(IDAYS1(I-1) .GT. IDAYS1(I))CALL ERRMES(4,15)                CALGEN
000180          GO TO 235                                                      CALGEN
000181      230 IF(IDAYS1(I-1) .GE. IDAYS1(I))CALL ERRMES(4,15)                CALGEN
000182      235 CONTINUE                                                       CALGEN
000183      255 IF(ISTC .EQ. 0)GO TO 400                                       CALGEN
000184          IPRNT = 0                                                      CALGEN
000185          DO 260 I=1,4                                                   CALGEN
000186          IF(DELTWE(I) .LT. -1.0 .OR. DELTWE(I) .GT. 1.0)CALL ERRMES(5,17) MODS
000187          IF(DELWEI(I) .LT. -1.0 .OR. DELWEI(I) .GT. 1.0)                MODS
000188        1 CALL ERRMES(5,17)                                              CALGEN
000189      260 CONTINUE                                                       CALGEN
000190      400 IF(GOOD1 .NE. 0)GO TO 430                                      CALGEN
000191          IF(GOOD2 .NE. 0)GO TO 420                                      CALGEN
000192          IF(FATAL .NE. 0)GO TO 420                                      MOD2
000193          NCROP = NCROP + 1                                              CALGEN
000194          IF(FATAL .EQ. 0)WRITE(ICALIN)ICTRY,IREG,IZONE,ISTRAT,ISUB,(IDAYS(ICALGEN
000195        1),I=1,8),(DELTWE(I),I=1,4),IPLNTE,ISEGSD,(IDAYS1(I),I=1,8),     CALGEN
000196        2(DELWEI(I),I=1,4),IPLNTI,ISGSD1                                 CALGEN
000197      420 IF(TLTST .EQ. 0)GO TO 450                                      CALGEN
000198          WRITE(IOUT,11)ICTRY,IREG,IZONE,ISTRAT,ISUB,ISW1,(IYPT(I),IMOT(I), MOD2
000199        1(DAT(I),I=1,8),TTYPE,ISW2,(IYR1(I),IMO1(I),IDA1(I),I=1,8),ITYPE1 MOD2
000200       11 FORMAT(6X,A4,2I4,2I5,A1,3(6I2,1X),6I2,I1/26X,A1,3(6I2,1X),6I2,I1) CALGEN
000201      430 IF(GOOD2 .NE. 0 .OR. TLTST .EQ. 0)GO TO 450                    CALGEN
000202          IF(ISTC .NE. 0)WRITE(IOUT,12)ICTRY2,IREG2,IZONE2,ISTRAT2,ISUB2, CALGEN
000203        1(DELTWE(I),I=1,4),IPLNTE,ISEGSD,(DELWEI(I),I=1,4),IPLNTI,ISGSD1, CALGEN
000204        2ITYPE2                                                          CALGEN
000205       12 FORMAT(6X,A4,2I4,2I5,2(4F5.3,I3,I2),2X,I1)                     MODS
000206      450 IREGL = IREG                                                   CALGEN
000207          IZONEL = IZONE                                                 CALGEN
000208          ISTRL = ISTRAT                                                 CALGEN
000209          ISUBL = ISUB                                                   MOD4
000210          GO TO 100                                                      CALGEN
000211      480 CALL ERRMES(4,2)                                               MOD4
000212      490 WRITE(ICALIN) ENDATA,(ZERO,I=1,32)                             CALGEN
000213          REWIND ICALIN                                                  CALGEN
000214          RETURN                                                         CALGEN
000215          END                                                            CALGEN
```

```
@ ELT ERRMES,1,760402, 49900    , 1

      SUBROUTINE ERRMES (ICARD,ICODE)                                   ERRMES
C        ERROR MESSAGE SUBROUTINE                                       ERRMES
C                                                                       ERRMES
      COMMON /CONST/                                                    CONST
     1PI,DEGTOR,       SLATMX,ENDATA,ZERO,INP,IOUT,INTSUR,INTSEG,ISIDF, CONST
     2ICWF,ISHF,LRSIDF,LRCWF,LRSHF,IWIN,MXSEG,ICALIN,ICWE,AREACF        CONST
     3,ISW1,ISW2,TGRDTP                                                 MOD8
      INTEGER ZERO                                                      CONST
      COMMON /FLAGS/                                                    FLAGS
     1IERR,IPRNT,GOOD1,NSHC,NSSC,NSFGC,NSUBR,NSEGR,NRSIDF,NRCWF,NRSHF,  FLAGS
     2ISSC,ISTAT,IREGL,IZONEL,ISTRL,ISUBL,ISEGL,NSEG,IMXSEG,NTLIST,     FLAGS
     3READSF,GOOD2,FATAL,IDAYS(8),IDSEG(150)                            FLAGS
     1,IMOI(8),IDAI(8),IYRI(8),IYR1(8),IMO1(8),IDA1(8)                  MOD2
      INTEGER GOOD1,GOOD2,READSF,FATAL                                  FLAGS
      COMMON /INPUT/                                                    INPUT
     1RSEED,SUBAR,HISTPW,DEVTPW,CV(3),DEVTPW,QLAT,QLONG,SLAT(3),SLONG(3)INPUT
     2,NAMSUB(2),ICTRY,ICTRY2,ICASE,ILIST,ITSEG,ISHC,ISSD,ISID,ISCW,    INPUT
     3ISEG,IREG,IZONE,ISTRAT,ISUB,ISW1,TYPE,ITYPE2,INDEX,ISEG2,IREG2,   INPUT
     4IZONE2,ISTRA2,ISUB2,LAT(3),LONG(3),IYR(8),IMO(8),IDA(8),CV4,IGRP, INPUT
     5DELTWF(4),IPLNIF,ISHCSD,ITRIND,NAGR,NAL,ITLIST(6),NCRCD,NCRFCD,   INPUT
     6NCRDR,IDAYS1(8),DELRE1(4),IPLNT1,ISGSD1,IDAS(8),IMOS(8),IYRS(8)   INPUT
      DOUBLE PRECISION RSEED                                            INPUT
      REAL NAMSUB,ICTRY,ICTRY2                                          INPUT
C                                                                       ERRMES
C                                                                       ERRMES
C                                                                       ERRMES
      IF ( IPRNT .NE. 0 )  GO TO 100                                    ERRMES
      WRITE (IOUT,800)                                                  ERRMES
  800 FORMAT (//24H ERROR IN THE FOLLOWING )                            ERRMES
      GO TO (10,20,30,40,50), ICARD                                     ERRMES
   10 WRITE (IOUT,801)  ICTRY,IREG,IZONE,ISTRAT,ISUB,NAMSUB,SUBAR,HISTPWERRMES
     1,NAGR,NAL,IGRP,ITYPE                                                      *NEW
      GOOD1= 1                                                          ERRMES  **-1
  801 FORMAT (31H SUBSTRATA HISTORICAL DATA CARD /                      ERRMES
     16X,A4,2I4,2I5,,2A4,F10.0,F7.3,2I4,I1,18X,I1/)                     MOD5
      IF ( ICODE .NE. 1 )  GO TO 90                                     ERRMES
      FATAL= 1                                                          ERRMES
   20 WRITE (IOUT,802) ICTRY2,IREG2,IZONE2,ISTRA2,ISUB2,DEVTPW,CV,DEVTPWERRMES
     1,CV4,ITYPE2                                                       MOD5
  802 FORMAT (32H SUBSTRATA STATISTICAL DATA CARD /                     ERRMES
     16X,A4,2I4,2I5,F6.3,5F5.3,2I2X,I1/)                                MOD5
      GOOD2= 1                                                          ERRMES
      GO TO 90                                                          ERRMES
   30 WRITE (IOUT,803)  ISEG,ICTRY,IREG,IZONE,ISTRAT,ISUB,LAT,LONG      ERRMES
     1  ,ISW,ITRIND,(ITLIST(I),I=1,6),ITYPE                             ERRMES
  803 FORMAT (28H SAMPLE SEGMENT ID DATA CARD /1X,I4                    ERRMES
     1,1X,A4,2I4,2I5,1X,A1,I2,1X,I2,1X,A1,I3,1X,I2,1X,A1,1X,I1,,1X,6I4  ERRMES
     1,8X,I1/)                                                          ERRMES
      GOOD1= 1                                                          ERRMES
      GO TO 90                                                          MOD2
   40 FATAL = 1                                                         MOD2
      WRITE(IOUT,804)ICTRY,IREG,IZONE,ISTRAT,ISUB,ISW1                  MOD2
     1,(IYRI(I),IMOI(I),IDAI(I),I=1,8)                                  MOD2
      WRITE(IOUT,804)ICTRY,IREG,IZONE,ISTRAT,ISUB,ISW2,                 MOD2
     1(IYR1(I),IMO1(I),IDA1(I),I=1,8)                                   MOD2
  804 FORMAT (31H 1 OR 2 CROP ANDON DATA CARDS /                        ERRMES
     1  1X       ,A4,2I4,2I5,A1,6I2,3(1X,6I2),I1/)                      ERRMES
      GOOD1= 1                                                          ERRMES
```

```
      IF(ICODE .NE. 2)GO TO 90                                          MOD2
   50 WRITE(IOUT,805)       ICTRY2,IREG2,I70NE2,ISTRA2,ISUB2,           ERRMES
     1(DELTWE(I),I=1,4),IPLNTF,ISEGSD,(DELWF1(I),I=1,4),IPLNT1,ISGSD1   ERRMES
      GOOD2=1                                                           ERRMES
  805 FORMAT(29H CROP WINDOW ERROR DATA CARDS /                         ERRMES
     11X,A4,1X,I3,I4,2I5,4F5.3,I3,I2,4F5.3,I3,I2)                       MOD5
   90 IPRNT= 1                                                          ERRMES
  100 IERR= IERR + 1                                                    ERRMES
      GO TO (110,120,130,140,150,160,170,180,190,200,210,220,230,240     ERRMES
     1  ,250,260,270,280,290), ICODE                                     ERRMES
  110 WRITE (IOUT,901)                                                  ERRMES
  901 FORMAT(55H INCONSISTENT SUBSTRATA HISTORICAL AND STATISTICAL DATA ERRMES
     1  /)                                                              ERRMES
      GO TO 999                                                         ERRMES
  120 WRITE (IOUT,902)                                                  ERRMES
  902 FORMAT(52H INCONSISTENT CROP WINDOW AND CROP WINDOW ERROR DATA/)  ERRMES
      GO TO 999                                                         ERRMES
  130 WRITE ( IOUT,903)                                                 ERRMES
  903 FORMAT(33H SUBSTRATA AREA MUST BE POSITIVE./)                     ERRMES
      GO TO 999                                                         ERRMES
  140 WRITE (IOUT,904)                                                  ERRMES
  904 FORMAT(51H HISTORICAL PROPORTION OF WHEAT MUST BE NONNEGATIVE/)   ERRMES
      GO TO 999                                                         ERRMES
  150 WRITE (IOUT,905)                                                  ERRMES
  905 FORMAT(45H DEVIPH MUST BE LESS THAN 9.999 IN MAGNITUDE./)         ERRMES
      GO TO 999                                                         ERRMES
  160 WRITE (IOUT,906)                                                  ERRMES
  906 FORMAT(51H COEFF. OF VARIATIONS MUST BE BETWEEN 0.0 AND 9.999/)   ERRMES
      GO TO 999                                                         ERRMES
  170 WRITE (IOUT,907)                                                  ERRMES
  907 FORMAT(37H DEVIPH MUST BE BETWEEN 0.0 AND 9.999/)                 ERRMES
      GO TO 999                                                         ERRMES
  180 WRITE (IOUT,908)                                                  ERRMES
  908 FORMAT(49H LATITUDE MUST BE LESS THAN OR EQUAL TO 65.0 DEG./)     ERRMES
      GO TO 999                                                         ERRMES
  190 WRITE (IOUT,909)                                                  ERRMES
  909 FORMAT(45H LONGITUDE MUST BE BETWEEN -180 AND +180 DEG./)         ERRMES
      GO TO 999                                                         ERRMES
  200 WRITE (IOUT,910)                                                  ERRMES
  910 FORMAT(46H SPRING/WINTER WHEAT INDICATOR MUST BE  W OR S/)        ERRMES
      GO TO 999                                                         ERRMES
  210 WRITE (IOUT,911)                                                  ERRMES
  911 FORMAT(34H CV4 MUST BE BETWEEN 0.0 AND 9.999/)                    MOD3
      GO TO 999                                                         ERRMES
  220 WRITE (IOUT,912)                                                  ERRMES
  912 FORMAT(29H YEAR MUST BE GREATER THAN 64/)                         ERRMES
      GO TO 999                                                         ERRMES
  230 WRITE (IOUT,913)                                                  ERRMES
  913 FORMAT(31H MONTH MUST BE BETWEEN 1 AND 12/)                       ERRMES
      GO TO 999                                                         ERRMES
  240 WRITE (IOUT,914)                                                  ERRMES
  914 FORMAT(29H DAY MUST BE BETWEEN 1 AND 31/)                         ERRMES
      GO TO 999                                                         ERRMES
  250 WRITE (IOUT,915)                                                  ERRMES
      GO TO 999                                                         ERRMES
  260 WRITE(IOUT,916)                                                   ERRMES
  916 FORMAT(35H GROUP NO MUST BE EQUAL TO 1,2 OR 3/)                   ERRMES
      GO TO 999                                                         ERRMES
  270 WRITE(IOUT,917)                                                   ERRMES
  917 FORMAT(67H DELTA ERRORS IN BIO WINDOW PREDICTION DATES ARE NOT BETERRMES
```

```
000119          IWELN-1 OR 1/)                                                  MODS
000120          GO TO 999                                                       ERRMES
000121      280 WRITE(IOUT,918)                                                 ERRMES
000122      918 FORMAT(73H SEGMENT SPECIFIED AS ORDINARY MUST HAVE A TRAINING SEGMERRMES
000123          1ENT PRIORITY LIST/)                                            ERRMES
000124          GO TO 999                                                       ERRMES
000125      290 WRITE(IOUT,919)                                                 ERRMES
000126      919 FORMAT(32H EITHER NAGR OR NAL CAN BE ZERO/)                     ERRMES
000127      915 FORMAT(25H OVERLAPPING CROP WINDOWS/)                           ERRMES
000128      999 RETURN                                                          ERRMES
000129          END                                                             ERRMES
```

```
* ELT FILGEN,1,760402, 49895   , 1

000001                SUBROUTINE FILGEN                                          FILGEN
000002          C        READS DATA FROM 4 INPUT FILES AND GENERATES 2 INTERMEDIATE  FILGEN
000003          C        DISK FILES.                                              FILGEN
000004          C        THE INPUT FILES ARE ...                                  FILGEN
000005          C           1. SUBSTRATA HISTORICAL DATA                          FILGEN
000006          C           2. SUBSTRATA STATISTICAL DATA                         FILGEN
000007          C           3. SAMPLE SEGMENT ID DATA                             FILGEN
000008          C           4. SAMPLE SEGMENT CROP WINDOW DATA                    FILGEN
000009          C                                                                 FILGEN
000010          C        THE OUTPUT FILES ARE ...                                 FILGEN
000011          C           1. SUBSTRATA DATA                                     FILGEN
000012          C           2. SAMPLE SEGMENT DATA                                FILGEN
000013          C                                                                 FILGEN
000014                COMMON /CONST/                                              CONST
000015               1PI,DEGTOR,      SLATMX,FNDATA,ZERO,INP,IOUT,INTSUB,INTSEG,ISIDF,  CONST
000016               2ICWF,ISHF,LRSIDF,LRCWF,LRSHF,ININ,MXSEG,ICALIN,ICWE,AREACF  CONST
000017               3,ISW1,ISW2,IGRDTP                                           MODR
000018                INTEGER ZERO                                                CONST
000019          C                                                                 FILGEN
000020                COMMON /FLAGS/                                              FLAGS
000021               1ERR,TPPNT,GOOD1,NSHC,NSSC,NSEGC,NSUBR,NSEGR,NRSIDF,NRLWF,NRSHF,  FLAGS
000022               2ISSC,ISTAT,IREGL,IZONEL,ISTRL,ISUBL,ISEGL,NSEG,IMXSEG,NTLIST,  FLAGS
000023               3READSF,GOOD2,FATAL,IDAYS(8),IDSEG(150)                      FLAGS
000024               1,IMOT(8),IDAT(8),IYRT(8),IYR1(8),IMO1(8),IDA1(8)            MOD2
000025                INTEGER GOOD1,GOOD2,READSF,FATAL                            FLAGS
000026          C                                                                 FILGEN
000027                COMMON /INPUT/                                              INPUT
000028               1RSEED,SUBSN,HTSTPN,DEVTPN,LV(3),DEVTPM,OLAT,OLONG,SLAT(3),SLONG(3)  INPUT
000029               2,NAMSUB(2),ICTRY,ICTRY2,ICASE,ILIST,ITSEG,ISHD,ISSD,ISID,ISCN,  INPUT
000030               3ISEG,IREG,IZONE,ISTRAT,ISUB,ISN,ITYPE,ITYPE2,INDEX,ISEG2,IREG2,  INPUT
000031               4IZONE2,ISTRA2,ISUB2,LAT(3),LONG(3),IYR(8),IMO(8),IDA(8),CV4,IGRP,  INPUT
000032               5DELTNE(8),IPLNTE,ISEGSD,ITRTNE,NACR,NAL,ITLIST(6),NCRCD,NCRECD,  INPUT
000033               6NCROP,IDAYS1(8),DELNE1(8),IPLNT1,ISGSD1,IDAS(8),IMOS(8),IYRS(8)  INPUT
000034                DOUBLE PRECISION RSEED                                      INPUT
000035                REAL NAMSUB,ICTRY,ICTRY2                                    INPUT
000036          C                                                                 FILGEN
000037                COMMON /LDATA/                                              LDATA
000038               1RWDATA(8),SWDATA(8)                                         LDATA
000039                INTEGER RWDATA,SWDATA                                       LDATA
000040                COMMON /IDATAB/                                             IDATAB
000041               1ITRAIN(2000),ITRPRI(2000)                                   IDATAB
000042          C                                                                 FILGEN
000043          C                                                                 FILGEN
000044          C                                                                 FILGEN
000045                NSHC= 0                                                     FILGEN
000046                NSSC= 0                                                     FILGEN
000047                NSEGC= 0                                                    FILGEN
000048                NSUBR= 0                                                    FILGEN
000049                NSEGR= 0                                                    FILGEN
000050                ISTAT= 0                                                    FILGEN
000051                IREGL= 0                                                    FILGEN
000052                IZONEL= 0                                                   FILGEN
000053                ISTRL= 0                                                    FILGEN
000054                ISUBL = 0                                                   MODR
000055                ISKSSC= 0                                                   FILGEN
000056                DO 5 I=1,2000                                               FILGEN
000057                ITRAIN(I) = 0                                               FILGEN
000058                ITRPRI(I) = 0                                               FILGEN
```

```
000059          5 CONTINUE                                                FILGEN
000060            ILOC = 1                                                FILGEN
000061            ILOC1 = 1                                               FILGEN
000062            IF ( ILIST .NE. 0 )  WRITE (IOUT,10)                    FILGEN
000063       10   FORMAT (44H1  SUBSTRATA HISTORICAL AND STATISTICAL DATA/) FILGEN
000064      C                                                             FILGEN
000065            IF ( ISHD .EQ. 0 )  GO TO 490                           FILGEN
000066      C        READ SUBSTRATA HISTORICAL DATA CARD                  FILGEN
000067       100  READ (ISHD,1)  ICTRY,IREG,IZONE,ISTRAT,ISUB,NAMSUB,SUBAR,HISTPN FILGEN
000068           1,NAGR,NAL,IGRP,ITYPE                                    MOD5
000069          1 FORMAT(5X,A4,2I4,2I5,2A4,F10.0,F7.5,2I4,I1,18X,I1)      MOD5
000070      C                                                             FILGEN
000071      C        TEST FOR END OF SUBSTRATA HISTORICAL DATA            FILGEN
000072      C                                                             FILGEN  **-1
000073       110  IF ( ICTRY .EQ. ENDATA )  GO TO 490                    FILGEN
000074            IF(IREG     IREGL)480,115,130                           MOD4
000075       115  IF(IZONE    - IZONEL)480,120,130                        MOD4
000076       120  IF(ISTRAT   -  ISTRL )480,125,130                       MOD4
000077       125  IF(ISUB    -  ISUBL)480,480,130                         MOD4
000078       130  CONTINUE                                                MOD4
000079            IF ( ICTRY .NE. 5HUSA .AND. ICTRY .NE. 4HCAND )  AREAGF= 0.01 FILGEN
000080            NSHC= NSHC + 1                                          FILGEN
000081            ISSC= 0                                                 FILGEN
000082            IF ( ISTAT .EQ. 0 )  GO TO 140                          FILGEN
000083      C        ISTAT .NE. 0.  LAST SUBSTRATUM STATISTICAL CARD READ HAD ZERO FILGEN
000084      C        FIELD FOR REGION, ZONE, STRATUM, OR SUBSTRATUM       FILGEN
000085            IF ( ISTAT .EQ. 4 )  GO TO 180                          FILGEN
000086            IF ( IREG .NE. IREGL )  GO TO 150                       FILGEN
000087      C        REGION FIELD OR SUBSTRATA HISTORICAL DATA CARD UNCHANGED. FILGEN
000088            IF ( ISTAT .EQ. 3 )  GO TO 180                          FILGEN
000089            IF ( IZONE .NE. IZONEL )  GO TO 150                     FILGEN
000090      C        ZONE FIELD UNCHANGED                                 FILGEN
000091            IF ( ISTAT .EQ. 2 )  GO TO 160                          FILGEN
000092            IF ( ISTRAT .EQ. ISTRL )  GO TO 180                     FILGEN
000093      C        IF ISKSSC .NE. 0, SKIP READING OF SUBSTRATA STATISTICAL DATA FILGEN
000094      C        CARD.                                                FILGEN
000095       140  IF ( ISKSSC .NE. 0 )  GO TO 180                         FILGEN
000096      C        READ NEXT SUBSTRATA STATISTICAL DATA CARD.           FILGEN
000097       150  READ (ISSD,2)  ICTRY2,IREG2,IZONE2,ISTRA2,ISUB2,DEVTPN,CV,DEVTPN FILGEN
000098           1,CV4,ITYPE2                                             MOD5
000099          2 FORMAT(5X,A4,2I4,2I5,F6.3,5F5.3,2I1X,I1)                MOD5
000100      C                                                             FILGEN
000101      C        TEST FOR END OF SUBSTRATA STATISTICAL DATA           FILGEN
000102       160  CONTINUE                                                FILGEN  **-1
000103      C                                                             FILGEN
000104            IF ( ICTRY2 .NE. ENDATA )  GO TO 170                    FILGEN
000105       165  WRITE (IOUT,21)                                         FILGEN
000106        21  FORMAT(//45H UNEXPECTED EOF ON SUBSTRATA STATISTICAL FILE) MOD4
000107            IERR= IERR + 1                                          FILGEN
000108            GO TO 490                                               FILGEN
000109       170  NSSC= NSSC + 1                                          FILGEN
000110            ISSC= 1                                                 FILGEN
000111      C        CHECK SUBSTRATA STATISTICAL DATA CARD FOR ZERO ZONE, STRATUM, FILGEN
000112      C        OR SUBSTRATUM DATA FIELDS AND SET ISTAT FLAG ACCORDINGLY. FILGEN
000113            ISTAT= 0                                                FILGEN
000114            IF ( ISUB2 .EQ. 0 )  ISTAT= 1                           FILGEN
000115            IF ( ISTRA2 .EQ. 0 )  ISTAT= 2                          FILGEN
000116            IF ( IZONE2 .EQ. 0 )  ISTAT= 3                          FILGEN
000117            IF ( IREG2 .EQ. 0 )  ISTAT= 4                           FILGEN
000118      C        CHECK FOR CONSISTENCY BETWEEN SUBSTRATA HISTORICAL DATA CARD FILGEN
```

```
000119        C         AND SUBSTRATA STATISTICAL DATA CARD.                    FILGEN
000120         180  GOOD1= 0                                                   FILGEN
000121              GOOD2= 0                                                   FILGEN
000122              IPRNT= 0                                                   FILGEN
000123              ISKSSC= 0                                                  FILGEN
000124              IF ( ICTRY .NE. ICTRY2 )  GO TO 190                        FILGEN
000125              IF ( ISTAT .EQ. 4 )  GO TO 200                             FILGEN
000126              IF ( IREG .NE. IREG2 )  GO TO 190                          FILGEN
000127              IF ( ISTAT .EQ. 3 )  GO TO 200                             FILGEN
000128              IF ( IZONE .NE. IZONE2 )  GO TO 190                        FILGEN
000129              IF ( ISTAT .EQ. 2 )  GO TO 200                             FILGEN
000130              IF ( ISTRAT .NE. ISTRA2 )  GO TO 190                       FILGEN
000131              IF ( ISTAT .EQ. 1 )  GO TO 200                             FILGEN
000132              IF ( ISUB .EQ. ISUB2 )  GO TO 200                          FILGEN
000133        C         SUBSTRATA HISTORICAL DATA AND SUBSTRATA STATISTICAL DATA FILGEN
000134        C         DO NOT AGREE                                           FILGEN
000135         190  CALL ERRMES (1,1)                                          FILGEN
000136        C         CHECK WHICH FIELD DISAGREED.                           FILGEN
000137              IF ( ISTAT .EQ. 4 )  GO TO 198                             FILGEN
000138              IF ( IREG - IREG2 )  198,192,150                           FILGEN
000139         192  IF ( ISTAT .EQ. 3 )  GO TO 198                             FILGEN
000140              IF ( IZONE - IZONE2 )  198,194,150                         FILGEN
000141         194  IF ( ISTAT .EQ. 2 )  GO TO 198                             FILGEN
000142              IF ( ISTRAT - ISTRA2 )  198,196,150                        FILGEN
000143         196  IF ( ISUB - ISUB2 )  198,100,150                           FILGEN
000144        C         SET ISKSSC TO READ SUBSTRATA HISTORICAL DATA CARD AND SKIP FILGEN
000145        C         READING SUBSTRATA STATISTICAL DATA CARD.               FILGEN
000146         198  ISKSSC= 1                                                  FILGEN
000147              GO TO 450                                                  FILGEN
000148        C         CHECK DATA ENTRIES ON SUBSTRATA HISTORICAL DATA CARD.  FILGEN
000149         200  IF ( SUBAR .GT. 0.0 )  GO TO 210                           FILGEN
000150              CALL ERRMES (1,3)                                          FILGEN
000151          210 IF(NAGR .LE. 0 .AND. NAL .NE. 0)GO TO 220                  MOD2
000152              CALL ERRMES(1,19)                                          FILGEN
000153          220 IF(IGRP .GT. 0 .AND. IGRP .LT. 4)GO TO 222                 MOD5
000154              CALL ERRMES(2,16)                                          MOD5
000155          222 IF(HISTPH .GE. 0.0 .AND. HISTPH .LE. 1.E2)GO TO 300        MOD5
000156              CALL ERRMES (1,4)                                          FILGEN
000157        C         IF SUBSTRATA STATISTICAL DATA CARD WAS JUST READ,      FILGEN
000158        C         CHECK DATA ENTRIES.  OTHERWISE SKIP OVER DATA CHECKING FILGEN
000159         300  IF ( ISSC .EQ. 0 )  GO TO 400                             FILGEN
000160              IPRNT= 0                                                   FILGEN
000161              IF(ABS(DEVIPH) .LE. 9.999)GO TO 217                        MOD5
000162              CALL ERRMES (2,5)                                          FILGEN
000163          217 IF(CV4 .GE. 0.0 .AND. CV4 .LE. 9.999)GO TO 320             MOD2
000164              CALL ERRMES(2,11)                                          MOD2
000165         320  DO 330 I=1,3                                               FILGEN
000166              IF ( CV(I) .GE. 0.0 .AND. CV(I) .LE. 9.999 )  GO TO 330    FILGEN
000167              CALL ERRMES (2,6)                                          FILGEN
000168         330  CONTINUE                                                   FILGEN
000169              IF ( DEVIPH .GE. 0.0 .AND. DEVIPH .LE. 9.999 )  GO TO 400  FILGEN
000170              CALL ERRMES (2,7)                                          FILGEN
000171        C                                                                FILGEN
000172         400  IF ( GOOD1 .NE. 0 )  GO TO 430                            FILGEN
000173              IF ( GOOD2 .NE. 0 )  GO TO 420                             FILGEN
000174              IF(FATAL .NE. 0)GO TO 420                                  MOD2
000175        C         ALL SUBSTRATA DATA IS VALID.  WRITE ONE RECORD OF SUBSTRATA FILGEN
000176        C         DATA ON INTERMEDIATE FILE.                             FILGEN
000177              NSUBR= NSUBR + 1                                           FILGEN
000178        C         CONVERT HISTPH TO STRAIGHT PERCENT                     FILGEN
```

```
000179  C         CONVERT UNITS OF AREA                                  FILGEN
000180        AREA= SUBAR*AREALF                                         FILGEN
000181        X = (NAM*10289.712)                                        MOD4
000182        RK = AREA/X                                                MOD4
000183        Y = (NAGR*RK*10289.712)                                    MOD4
000184        HSTPW = (HISTPW*AREA)/Y                                    MOD4
000185        IF ( FATAL .EQ. 0 )                                        FILGEN
000186       1  WRITE (INTSUB)  ICTRY,IREG,IZONE,ISTRAT,ISUB,NAMSUB,AREA,HSTPW  FILGEN
000187       1       ,DEVTPW,CV,DEVTPM,NAGR,NAL,CV4,IGRP                 FILGEN
000188  420  IF ( TLIST .EQ. 0 )  GO TO 450                             FILGEN
000189       WRITE (IOUT,11)  ICTRY,IREG,IZONE,ISTRAT,ISUB,NAMSUB,SUBAR,HISTPW  FILGEN
000190      1,NAGR,NAL,IGRP,ITYPE                                       MOD5
000191   11 FORMAT(6X,A4,2I4,2I5,2A4,F10.0,F7.3,2I4,I1,18X,I1)          MOD5
000192  430  IF ( GOOD2 .NE. 0 )  GO TO 450                             FILGEN
000193       IF ( ILIST .EQ. 0 )  GO TO 450                             FILGEN
000194       IF ( ISSC .NE. 0 )  WRITE (IOUT,12)  ICTRY2,IREG2,IZONE2,ISTRA2  FILGEN
000195      1,ISUB2,DEVTPW,CV,DEVTPM,CV4,ITYPE2                         MOD5
000196   12 FORMAT(6X,A4,2I4,2I5,F6.3,5F5.3,21X,I1)                     MOD5
000197  450  IREGL= IREG                                                FILGEN
000198       IZONEL= IZONE                                              FILGEN
000199       ISTRL= ISTRAT                                              FILGEN
000200       ISUBL = ISUB                                               MOD4
000201       GO TO 100                                                  FILGEN
000202  C                                                               FILGEN
000203  480 CALL ERRLFS(1,1)                                            MOD4
000204  C         WRITE TRAILER RECORD ON INTERMEDIATE SUBSTRATA FILE   FILGEN
000205  490  WRITE (INTSUB)  ENDATA, ( ZERO,I=1,17 )                    FILGEN
000206       REWIND INTSUB                                              FILGEN
000207       CALL CALGEN                                                FILGEN
000208       IF ( ISID .EQ. 0 )  GO TO 900                              FILGEN
000209       IMXSLG= 0                                                  FILGEN
000210       NSEG= 0                                                    FILGEN
000211       IREGL= 0                                                   FILGEN
000212       IZONEL = 0                                                 FILGEN
000213       ISTRL= 0                                                   FILGEN
000214       ISUBL= 0                                                   FILGEN
000215       ISEGL = 0                                                  FILGEN
000216       ISKCH= 0                                                   MOD4
000217       IF ( ILIST .NE. 0 )  WRITE (IOUT,19)                       FILGEN
000218   19  FORMAT (20H1  SAMPLE SEGMENT ID/)                          FILGEN
000219  500  READ (ISID,3)  ISEG,ICTRY,IREG,IZONE,ISTRAT,ISUB,LAT,LONG  FILGEN
000220      1  ,TSP,ITRIND,(ITLIST(I),I=1,6),ITYPE                      FILGEN
000221    3  FORMAT (I4,1X,A4,2I4,2I5,1X,A1,I2,1X,I2,1X,A1,I3,1X,I2,1X,A1  FILGEN
000222      1  ,1X,I1,1X,6I4,8X,I1)                                     FILGEN
000223  C                                                               FILGEN
000224  C         TEST FOR END OF SEGMENT ID DATA                       FILGEN
000225  C                                                               FILGEN
000226  510  IF ( ICTRY .EQ. ENDATA )  GO TO 900                        FILGEN
000227       IF(IREG    -  IREGL)997,503,515                            MOD4
000228   503 IF(IZONE   -  IZONEL)997,505,515                           MOD4
000229   505 IF(ISTRAT  -  ISTRL)997,507,515                            MOD4
000230   507 IF(ISUB    -  ISUBL)997,512,515                            MOD4
000231   512 IF(ISEG    -  ISEGL)997,997,515                            MOD4
000232   515 CONTINUE                                                   FILGEN
000233       NSEGL= NSEGL + 1                                           FILGEN
000234       SLAT(2)= LAT(2)                                            FILGEN
000235       SLAT(3)= LAT(3)                                            FILGEN
000236       SLONG(2)= LONG(2)                                          FILGEN
000237       SLONG(3)= LONG(3)                                          FILGEN
000238  530  GOOD1= 0                                                   FILGEN
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
000239  C                                                                       FILGEN
000240  C        CHECK DATA ENTRIES ON SAMPLE SEGMENT ID DATA CARD.             FILGEN
000241  C        CONVERT LATITUDE MIN. TO DEGREES AND ADD TO LATITUDE DEGREES.  FILGEN
000242   600  QLAT=   SLAT(2) + SLAT(3)/60.0                                    FILGEN
000243        IF ( QLAT .LE. SLATMX )  GO TO 610                                FILGEN
000244        CALL ERRMES (3,8)                                                 FILGEN
000245   610  IF ( LAT(1) .EQ. INS )  QLAT= - QLAT                              FILGEN
000246  C        CONVERT LONGITUDE MIN. TO DEGREES AND ADD TO LONGITUDE DEGREES.FILGEN
000247        QLONG= SLONG(2) + SLONG(3)/60.0                                   FILGEN
000248        IF ( QLONG .LE. 180.0 )  GO TO 620                                FILGEN
000249        CALL ERRMES (3,9)                                                 FILGEN
000250  C        FOR NEGATIVE (WESTERN) LONGITUDES SUBTRACT MAGITUDE FROM 360.  FILGEN
000251   620  IF ( LONG(1) .EQ. IRW )  QLONG= 360.0 - QLONG                     FILGEN
000252  C        CONVERT LATITUDE AND LONGITUDE TO RADIANS                      FILGEN
000253        QLAT= QLAT*DEGTOR                                                 FILGEN
000254        QLONG= QLONG*DEGTOR                                               FILGEN
000255  C        CHECK SPRING/WINTER WHEAT INDICATOR.                           FILGEN
000256        IF ( ISW .NE. IPS .AND. ISW .NE. IRW )  CALL ERRMES (3,10)        FILGEN
000257        IF(ITSEG .EQ. 0)ITRIND = 0                                        FILGEN
000258        IF(ITSEG .EQ. 0)GO TO 800                                         FILGEN
000259        IF(ITRIND .EQ. 0)GO TO 650                                        FILGEN
000260        ICT = 0                                                           FILGEN
000261        DO 625 I=1,6                                                      FILGEN
000262        IF(ITLIST(I) .EQ. 0)GO TO 625                                     FILGEN
000263        ICT = ICT + 1                                                     FILGEN
000264   625  CONTINUE                                                          FILGEN
000265        IF(ICT .NE. 0)GO TO 627                                           FILGEN
000266        CALL ERRMES(3,18)                                                 FILGEN
000267   627  DO 640 I=1,ICT                                                    FILGEN
000268        DO 630 J=1,2000                                                   FILGEN
000269        IF(J .EQ. ILOC1)GO TO 635                                         FILGEN
000270        IF(ITLIST(I) .EQ. ITRPRT(J))GO TO 640                             FILGEN
000271   630  CONTINUE                                                          FILGEN
000272   635  ITRPRT(ILOC1) = ITLIST(I)                                         FILGEN
000273        ILOC1 = ILOC1 + 1                                                 FILGEN
000274   640  CONTINUE                                                          FILGEN
000275        GO TO 800                                                         FILGEN
000276   650  ITRAIN(ILOC) = ISEG                                               FILGEN
000277        ILOC = ILOC + 1                                                   FILGEN
000278        IF(ILOC .LE. 7000 .AND. ILOC1 .LE. 2000)GO TO 800                 FILGEN
000279        WRITE(IOUT,18)ILOC,ILOC1                                          FILGEN
000280        GO TO 1000                                                        FILGEN
000281  C                                                                       FILGEN
000282   800  IF ( GOOD1 .NE. 0 )  GO TO 830                                    FILGEN
000283  C        ALL SEGMENT SAMPLE DATA IS GOOD.  WRITE ONE RECORD OF SEGMENT  FILGEN
000284  C        INTERMEDIATE FILE.                                             FILGEN
000285        IF(FATAL .EQ. 0)NSEGR = NSEGR + 1                                 MOD2
000286  C        COUNT NO. OF SEGMENTS IN THIS SUBSTRATUM.                      FILGEN
000287        IF ( ISUB .NE. ISUBL )  GO TO 810                                 FILGEN
000288        IF ( ISTRAT .NE. ISTRL )  GO TO 810                               FILGEN
000289        IF ( IZONE .NE. IZONEL )  GO TO 810                               FILGEN
000290        IF ( IREG .EQ. IREGL )  GO TO 820                                 FILGEN
000291  C        NEW SUBSTRATUM. ADJUST IMXSEG AND REINITIALIZE COUNTERS.       FILGEN
000292   810  IMXSEG= MAX0 (IMXSEG,NSEG)                                        FILGEN
000293        NSEG= 0                                                           FILGEN
000294        IREGL= IREG                                                       FILGEN
000295        IZONEL= IZONE                                                     FILGEN
000296        ISTRL= ISTRAT                                                     FILGEN
000297        ISUBL= ISUB                                                       FILGEN
000298   820  NSEG= NSEG + 1                                                    FILGEN
```

```
      ISEGL = ISEG                                                      MOD4
      IF ( FATAL .EQ. 0 )                                               FTLGEN
     1  WRITE (INTSEG)  ICTRY,ISEG,IREG,IZONE,ISTRAT,ISUB,GLAT,GLONG   FTLGEN
     1  ,ISW,ITRIND,(ITLIST(I),I=1,6)                                  FTLGEN
  830 IF ( ILIST .EQ. 0 )  GO TO 500                                    FTLGEN
C        IF ILIST .NE. 0, PRINT OUT CARD IMAGES FOR SAMPLE SEGMENT DATA.FTLGEN
      WRITE (IOUT,13)  ISEG,ICTRY,IREG,IZONE,ISTRAT,ISUB,LAT,LONG      FTLGEN
     1  ,ISW,ITRIND,(ITLIST(I),I=1,6),ITYPE                            FTLGEN
   13 FORMAT (1X,I4,1X,A4,2I4,2I5,1X,A1,I2,1X,I2,1X,A1,I3,1X,I2,1X,A1  FTLGEN
     1,1X,I1,1X,6I4,8X,I1)                                              FTLGEN
      GO TO 500                                                         FTLGEN
C                                                                       FTLGEN
C        WRITE TRAILER RECORD ON INTERMEDIATE SEGMENT FILE              FTLGEN
  900 WRITE (INTSEG)  ENDATA, ( ZERO,I=1,15)                            FTLGEN
      REWIND INTSEG                                                     FTLGEN
      IMXSEG= MAXO (IMXSEG,NSEG)                                        FTLGEN
      IF ( IMXSEG .LE. MXSEG )  GO TO 990                               FTLGEN
      WRITE (IOUT,23)  IMXSEG,MXSEG                                     FTLGEN
   23 FORMAT (//40H MAX. NO. OF SEGMENTS IN ANY SUBSTRATA =I5           FTLGEN
     1  ,19H  EXCEEDING MXSEG =I5)                                      FTLGEN
 1000 STOP                                                              FTLGEN
  990 IF(IISEG .EQ. 0)GO TO 999                                         FTLGEN
      DO 998 I=1,ILOC1                                                  FTLGEN
      ISEGPR =.ITMPRI(I)                                                FTLGEN
      DO 995 J=1,ILOC                                                   FTLGEN
      IF(ITPATN(J) .EQ. ISEGPR)GO TO 998                                FTLGEN
  995 CONTINUE                                                          FTLGEN
      WRITE(IOUT,20)ISEGPR                                              FTLGEN
   20 FORMAT(//27H TRAINING PRIORITY SEGMENT ,I4,48H IS EITHER NOT A TRAFTLGEN
     1INING SEGMENT OR NONEXISTANT)                                     FTLGEN
      IERR = IERR + 1                                                   FTLGEN
      FATAL = 1                                                         FTLGEN
  998 CONTINUE                                                          FTLGEN
      GO TO 999                                                         MOD4
  997 FATAL = 1                                                         MOD4
      IERR = IERR + 1                                                   MOD4
      WRITE(IOUT,25)IREG,IZONE,ISTRAT,ISUB,ISEG                         MOD4
   25 FORMAT(1H0,45H SAMPLE SEGMENT ID CARD IS OUT OF ORDER. ID = ,     MOD4
     1 2I4,3I5)                                                         MOD4
  999 WRITE (IOUT,17)  NSHC,NSSC,NSEGC,NCRCD,NCRECD,NSUBR,NSEGR,        FTLGEN
     1NCROP,IERR                                                        FTLGEN
   17 FORMAT (//1X,I4,38H  SUBSTRATA HISTORICAL DATA CARDS READ         FTLGEN
     1  /1X,I4,39H  SUBSTRATA STATISTICAL DATA CARDS READ               FTLGEN
     2  /1X,I4,35H  SAMPLE SEGMENT ID DATA CARDS READ                   FTLGEN
     2/1X,I4,31H  CROP CALENDER DATA CARDS READ                         FTLGEN
     2/1X,I4,37H  CROP CALENDER ERROR DATA CARDS READ                   FTLGEN
     3  /1X,I4,45H  SUBSTRATA INTERMEDIATE FILE RECORDS WRITTEN         FTLGEN
     4  /1X,I4,43H  SEGMENT INTERMEDIATE FILE RECORDS WRITTEN           FTLGEN
     2/1X,I4,49H  CROP CALENDER INTERMEDIATE FILE RECORDS WRITTEN       FTLGEN
     5  ///1X,I4,28H   INPUT ERRORS DETECTED)                           FTLGEN
C                                                                       FTLGEN
      IF ( FATAL .EQ. 0 )  RETURN                                       FTLGEN
      WRITE (IOUT,24)                                                   FTLGEN
   24 FORMAT (//36H   FATAL ERROR IN INPUT. JOB ABORTED)                FTLGEN
   18 FORMAT(//44H MAX NO OF TRAINING SEGMENTS IS 2000.ILOC = ,I5,9H ILOFTLGEN
     1C1 = ,I5)                                                         FTLGEN
      STOP                                                              FTLGEN
C                                                                       FTLGEN
      END                                                               FTLGEN
```

```
@ ELT GRDCAL,1,760401, 67418   , 1

000001          SUBROUTINE GRDCAL(IGRID,JGRID,INDEX,FI,FJ,IFR)                    GRDCAL
000002    C     THIS ROUTINE COMPUTES THE BEST INDEX VALUE FOR A GIVEN GRID PAIR  GRDCAL
000003    C     OF COORDINATES IN THE GLOBAL STEREOGRAPHIC INDEX MATRIX           GRDCAL
000004          COMMON /CONST/                                                    CONST
000005         1PI,DEGRAD,       SLATMX,ENDATA,ZERO,INP,IOUT,INISUB,INTSEG,ISIDE, CONST
000006         2ICHE,ISHE,LRSIDE,LRCHE,LRSHE,IBIN,MXSEG,ICALIN,ICHE,AREACE       CONST
000007         3,ISW1,ISW2,IGRDTP                                                 MODB
000008          INTEGER ZERO                                                      CONST
000009          COMMON /DIRAC/                                                    DIRAC
000010         1ICOUN(252),IREC(500)                                              DIRAC
000011          DIMENSION ISTATE(8,8),IBOUND(2,8),IPI(8),ISTAT(64),IBOUN(16)      GRDCAL
000012          EQUIVALENCE(ISTATE(1,1),ISTAT(1)),(IBOUND(1,1),IBOUN(1))          GRDCAL
000013          DATA ISTAT/1,3,2,7,8,5,4,6,3,1,2,5,4,7,8,6,3,1,4,5,2,7,6,8,5,3,4,7GRDCAL
000014         1,6,1,2,8,5,7,6,3,4,1,8,2,7,5,6,1,8,3,4,2,7,1,8,5,6,3,2,4,1,7,8,3,2GRDCAL
000015         2,5,6,4/                                                           GRDCAL
000016          DATA IBOUN/1,0,1,1,0,1,-1,1,-1,0,-1,-1,0,-1,1,-1/                  GRDCAL
000017          CALL RDGRID(JGRID)                                                GRDCAL
000018          JGSAV = JGRID                                                     GRDCAL
000019          IF(IREC(IGRID) .LO. 0)GO TO 10                                    GRDCAL
000020          INDEX = IREC(IGRID)                                               GRDCAL
000021        5 RETURN                                                            GRDCAL
000022       10 Y = FJ - JGRID                                                    GRDCAL
000023          X = FI - IGRID                                                    GRDCAL
000024          IF(Y .NE. 0. .OR. X .NE. 0.)GO TO 15                              MOD9
000025          ISTA = 1                                                          MOD9
000026          GO TO 50                                                          MOD9
000027       15 IF(Y .EQ. 0.)X = X + .0000001                                     MOD9
000028          SLOPE = Y/X                                                       GRDCAL
000029          IF(X .GT. 0. .AND. Y .GE. 0.)GO TO 40                             GRDCAL
000030          IF(X .LT. 0. .AND. Y .GE. 0.)GO TO 30                             GRDCAL
000031          IF(X .LT. 0. .AND. Y .LE. 0.)GO TO 20                             GRDCAL
000032          IF(SLOPE .LT. -1.)ISTA = 7                                        MOD9
000033          IF(SLOPE .GE. -1.)ISTA = 8                                        MOD9
000034          GO TO 50                                                          GRDCAL
000035       20 IF(SLOPE .LT. 1.)ISTA = 5                                         MOD9
000036          IF(SLOPE .GE. 1.)ISTA = 6                                         MOD9
000037          GO TO 50                                                          GRDCAL
000038       30 IF(SLOPE .LT. -1.)ISTA = 3                                        MOD9
000039          IF(SLOPE .GE. -1.)ISTA = 4                                        MOD9
000040          GO TO 50                                                          GRDCAL
000041       40 IF(SLOPE .GE. 1.)ISTA = 2                                         MOD9
000042          IF(SLOPE .LT. 1.)ISTA = 1                                         MOD9
000043       50 DO 55 I=1,8                                                       GRDCAL
000044          IPI(I) = ISTATE(I,ISTA)                                           MOD9
000045       55 CONTINUE                                                          GRDCAL
000046          DO 75 ICT=1,8                                                     GRDCAL
000047          IPIT = IPI(ICT)                                                   GRDCAL
000048          IG = IGRID + IBOUND(1,IPIT)                                       GRDCAL
000049          JG = JGRID + IBOUND(2,IPIT)                                       GRDCAL
000050          IF(JG .EQ. JGSAV)GO TO 65                                         GRDCAL
000051          CALL RDGRID(JG)                                                   GRDCAL
000052          JGSAV = JG                                                        GRDCAL
000053       65 IF(IREC(IG) .EQ. 0)GO TO 75                                       GRDCAL
000054          INDEX = IREC(IG)                                                  GRDCAL
000055          GO TO 5                                                           GRDCAL
000056       75 CONTINUE                                                          GRDCAL
000057       80 IFR = 1                                                           GRDCAL
000058          GO TO 5                                                           GRDCAL
```

```
* FLT GRID,1,760401, 67416      , 1

000001                SUBROUTINE GRID(IERT)                                    GRID
000002          C     THIS SUBROUTINE CONTROLS THE CALCULATION OF THE INDEX NO GRID
000003          C     ASSOCIATED WITH A GIVEN LATITUDE AND LONGITUDE.(FOR LUMP) GRID
000004                COMMON /CONST/                                           CONST
000005               1PI,DEGTOR,       SLATMX,ENDATA,ZERO,INP,IOUT,INTSUB,INTSEG,ISIDF,  CONST
000006               2ICWF,ISHF,LRSIDF,LRCWF,LRSHF,IHTN,MXSEG,ICALIN,ICWE,AREALF  CONST
000007               3,IS41,IS42,IGRDTP                                          MODA
000008                INTEGER ZERO                                             CONST
000009                COMMON /INPUT/                                           INPUT
000010               1KSEED,SUBAR,HISTPW,DEVTPW,CV(3),DEVTPM,OLAT,OLONG,SLAT(3),SLONG(3)INPUT
000011               2,NAMSUB(2),ICTRY,ICTRY2,ICASE,ILIST,ITSEG,ISND,ISSD,ISID,ISCW,  INPUT
000012               3ISEG,IREG,IZONE,ISTRAT,ISUB,ISW,ITYPE,ITYPE2,INDEX,ISEG2,IREG2,  INPUT
000013               4IZONE2,ISTRA2,ISUB2,LAT(3),LONG(3),IYR(8),IMO(8),IDA(8),CV4,IGRP,  INPUT
000014               5DELTHE(8),IPLNHE,TSEGSD,ITRTND,NAGR,NAL,ITLIST(6),NCRCD,NCRECD,  INPUT
000015               6NERDP,IDAYSM(8),DELSE1(8),IPLNT1,ISGSD1,IDAS(8),IMOS(8),IYRS(8)  INPUT
000016                DOUBLE PRECISION RSEED                                   INPUT
000017                REAL NAMSUB,ICTRY,ICTRY2                                 INPUT
000018                DATA I0,J0,PI/125,125,3.1415926/                         GRID
000019                IERT = 0                                                 GRID
000020                PHIP = ABS(OLAT)                                         GRID
000021                X =(PI -2.*PHIP)/4.                                      GRID
000022                R = 120.8174940*TAN(X)                                   GRID
000023                A = OLONG - PI/18.                                       GRID
000024                FI = I0 + R*COS(A)                                       GRID
000025                FJ = J0 - R*SIN(A)                                       GRID
000026                IF(OLAT .LT. 0.)FI = FI +2.*I0                           GRID
000027                IGRID = FI + .5                                          GRID
000028                JGRID = FJ + .5                                          GRID
000029                CALL GRDCAL(IGRID,JGRID,INDEX,FI,FJ,IERT)                GRID
000030                IF(IERT .EQ. 0)GO TO 100                                 GRID
000031                WRITE(IOUT,900)IREG,IZONE,ISTRAT,ISUB,ISEG               GRID
000032            100 RETURN                                                   GRID
000033            900 FORMAT(1H0, 8H REGION-,I3,6H ZONE-,I3,8H STRATA-,I4,11H SUBSTRATA-GRID
000034               1,I4,9H SEGMENT-,I4/1H ,73H NO WEATHER GRID INDEX NO. COULD BE FOUNGRID
000035               2D. THE SEGMENT RECORD IS SKIPPED.)                       GRID
000036                END                                                      GRID
```

```
• ELT IBETAI•1•760405• 55991    • 1

000001          SUBROUTINE IBETAI(X•A•B•P•IER)                                  IBETAI
000002     C                                                                    IBETAI
000003     C  COMPUTE INCOMPLETE BETA INTEGRAL FOR ARGUMENTS                    IBETAI
000004     C   X BETWEEN ZERO AND ONE• A AND B POSITIVE.                        IBETAI
000005     C                                                                    IBETAI
000006     C   X  VALUE TO WHICH FUNCTION IS TO BE INTEGRATED                   IBETAI
000007     C   A  FIRST INPUT PARAMETER                                         IBETAI
000008     C   B  SECOND INPUT PARAMETER                                        IBETAI
000009     C   P  OUTPUT PROBABILITY THAT A RANDOM VARIABLE FROM A              IBETAI
000010     C          BETA DISTRIBUTION HAVING PARAMETERS A AND B               IBETAI
000011     C           WILL BE LESS THAN OR EQUAL TO X                          IBETAI
000012     C   IER  ERROR FLAG WITH POSSIBLE VALUES                             IBETAI
000013     C       =40 X NOT BETWEEN RANGE 0 TO 1                               IBETAI
000014     C       =50  A AND/OR B NOT POSITIVE                                 IBETAI
000015     C       =60  GAMMA FUNCTION RANGE VIOLATED (NOT .GT. 0. AND .LE. 88.) IBETAI
000016     C                                                                    IBETAI
000017          LOGICAL INDEX                                                   IBETAI
000018     C   TEST FOR ADMISSIBILITY OF ARGUMENTS                              IBETAI
000019          DATA ACU /1.E-6/                                                IBETAI
000020          P=X                                                             IBETAI
000021          IER=50                                                          IBETAI
000022          IF(A.LE.0.  .OR. B.LE.0.) RETURN                                IBETAI
000023          IER=40                                                          IBETAI
000024          IF(X.LE.0.  .OR. X.GE.1.) RETURN                                IBETAI
000025          IER=0                                                           IBETAI
000026     C                                                                    IBETAI
000027     C   CHANGE TAIL IF NECESSARY AND DETERMINE S                         IBETAI
000028          ASB=A+B                                                         IBETAI
000029          CX=1.0-X                                                        IBETAI
000030          IF(A.GE.ASB*X) GO TO 10                                         IBETAI
000031          XX=CX                                                           IBETAI
000032          CX=X                                                            IBETAI
000033          AA=B                                                            IBETAI
000034          BB=A                                                            IBETAI
000035          INDEX=.TRUE.                                                    IBETAI
000036          GO TO 20                                                        IBETAI
000037       10 XX=X                                                            IBETAI
000038          AA=A                                                            IBETAI
000039          BB=B                                                            IBETAI
000040          INDEX = .FALSE.                                                 IBETAI  *NEW
000041       20 TERM=1.0                                                        IBETAI  **-1
000042          AT=1.0                                                          IBETAI
000043          P=1.0                                                           IBETAI
000044          NS=BB+CX*ASB                                                    IBETAI
000045     C                                                                    IBETAI
000046     C   USE SOPER REDUCTION FORMULAE                                     IBETAI
000047          RX=XX/CX                                                        IBETAI
000048       30 TEMP=BB-AT                                                      IBETAI
000049          IF(NS.LE.0.0) RX=XX                                             IBETAI
000050       40 TERM=TERM*TEMP*RX/(AA+AT)                                       IBETAI
000051          P=P+TERM                                                        IBETAI
000052          TEMP=ABS(TERM)                                                  IBETAI
000053          IF(TEMP.LE.ACU.AND.TEMP.LE.ACU*P) GO TO 50                      IBETAI
000054          AT=AT+1.0                                                       IBETAI
000055          NS=NS-1                                                         IBETAI
000056          IF(NS.GT.0) GO TO 30                                            IBETAI
000057          TEMP=ASB                                                        IBETAI
000058          ASB=ASB+1.0                                                     IBETAI
```

```
000059          GO TO 40                                                IBETAI
000060    C                                                             IBETAI
000061    C   CALCULATE RESULT USING GAMMA ROUTINE                      IBETAI
000062       50 CONTINUE                                                IBETAI
000063          CC=AA+BB                                                IBETAI
000064          CALL ALGAMA(AA,Y,IER)                                   IBETAI
000065          IF(IER.GT.0) GO TO 60                                   IBETAI
000066          CALL ALGAMA(BB,Z,IER)                                   IBETAI
000067          IF(IER.GT.0) GO TO 60                                   IBETAI
000068          CALL ALGAMA(CC,W,IER)                                   IBETAI
000069          IF(IER.GT.0) GO TO 60                                   IBETAI
000070          BETA=EXP(Y+Z-W)                                         IBETAI
000071          P=P+EXP(AA*ALOG(XX)+(BB-1.0)*ALOG(CX))/(AA*BETA)        IBETAI
000072          IF(TOPEX)P=1.0-P                                        IBETAI
000073          RETURN                                                  IBETAI
000074    C   ERROR RETURN                                              IBETAI
000075       60 CONTINUE                                                IBETAI
000076          P=0.                                                    IBETAI
000077          RETURN                                                  IBETAI
000078          END                                                     IBETAI
```

```
• ELT LFPA,1,760401, 67410      , 1

000001                SUBROUTINE LFPA (FLDA,LMO,LYR,DAYS)                    LFPA
000002          C                                                            LFPA
000003          C MODULE - P1LFPA                                            LFPA
000004          C AUGUST 27, 1975                                            LFPA
000005          C                                                            LFPA
000006          C                                                            LFPA
000007          C  JUNE 26,1973                                              LFPA
000008          C                                                            LFPA
000009          C  MODULE PPLFPA - POINT TARGET PERFORMANCE PREDICTOR        LFPA
000010          C                                                            LFPA
000011          C  GIVEN  DAY,MONTH,YEAR - SUBR. LFPA RETURNS THE RIGHT ASCENSION OF  LFPA
000012          C  GREENWICH AT MIDNIGHT OF A GIVEN DAY                      LFPA
000013          C                                                            LFPA
000014          C     REFERENCE EPOCH IS 0 HOUR 1JAN1950                     LFPA
000015          C                                                            LFPA
000016          C     COMPUTE DAYS IN FULL YEARS FROM EPOCH TO LYR           LFPA
000017          C     DAYS TO 1JAN1963 IS 4748.                              LFPA
000018          C                                                            LFPA
000019                DAYS=4748.                                             LFPA
000020          C     NOTE--LYR MUST BE GREATER THAN OR EQUAL TO (1)64       LFPA
000021                LASTYR=LYR-1                                           LFPA
000022                DO 20 I=63,LASTYR                                      LFPA
000023                KREMD=MOD((I-60),4)                                    LFPA
000024                IF(KREMD.GT.0) GO TO 10                                LFPA
000025                DAYS=DAYS+366                                          LFPA
000026                GO TO 20                                               LFPA
000027             10 DAYS=DAYS+365                                          LFPA
000028             20 CONTINUE                                               LFPA
000029                IF(LMO-1)30,40,30                                      LFPA
000030             30 IF(LMO-2)60,50,60                                      LFPA
000031             40 DAYS=DAYS+FLDA-1.                                      LFPA
000032                GO TO 270                                              LFPA
000033             50 DAYS=DAYS+FLDA+30.                                     LFPA
000034                GO TO 270                                              LFPA
000035             60 KDEL=MOD((LYR-60),4)                                   LFPA
000036                IF(KDEL.GT.0) GO TO 70                                 LFPA
000037                DAYS=DAYS+59.                                          LFPA
000038                GO TO 80                                               LFPA
000039             70 DAYS=DAYS+58.                                          LFPA
000040             80 IF(LMO-3) 270,170,90                                   LFPA
000041             90 IF(LMO-5) 180,190,100                                  LFPA
000042            100 IF(LMO-7) 200,210,110                                  LFPA
000043            110 IF(LMO-9) 220,230,120                                  LFPA
000044            120 IF(LMO-11)240,250,260                                  LFPA
000045            170 DAYS=DAYS+FLDA                                         LFPA
000046                GO TO 270                                              LFPA
000047            180 DAYS=DAYS+FLDA+31.                                     LFPA
000048                GO TO 270                                              LFPA
000049            190 DAYS=DAYS+FLDA+61.                                     LFPA
000050                GO TO 270                                              LFPA
000051            200 DAYS=DAYS+FLDA+92.                                     LFPA
000052                GO TO 270                                              LFPA
000053            210 DAYS=DAYS+FLDA+122.                                    LFPA
000054                GO TO 270                                              LFPA
000055            220 DAYS=DAYS+FLDA+153.                                    LFPA
000056                GO TO 270                                              LFPA
000057            230 DAYS=DAYS+FLDA+184.                                    LFPA
000058                GO TO 270                                              LFPA
```

```
000059          240 DAYS=DAYS+FLDA+214.                                  LFPA
000060              GO TO 270                                            LFPA
000061          250 DAYS=DAYS+FLDA+245.                                  LFPA
000062              GO TO 270                                            LFPA
000063          260 DAYS=DAYS+FLDA+275.                                  LFPA
000064          270 CONTINUE                                             LFPA
000065        C                                                          LFPA
000066              RETURN                                               LFPA
000067        C                                                          LFPA
000068              END                                                  LFPA
```

```
@ ELT LUMP,1,760402, 49896      , 1

000001                SUBROUTINE LUMP                                          LUMP
000002          C        GENERATES OUTPUT FILES                                LUMP
000003          C          (SEGMENT ID FILE, CROP WINDOW FILE, AND SUBSTRATA HISTORICAL LUMP
000004          C            FILE)                                              LUMP
000005          C                                                               LUMP
000006                COMMON /CONST/                                            CONST
000007               1PI,DEGTOR,      SLATMX,ENDATA,ZERO,INP,IOUT,INTSUR,INTSEG,ISIDF, CONST
000008               2ICWF,ISHF,LRSIDF,LRCWF,LRSHF,IWIN,MXSEG,ICALIN,ICWE,AREALF  CONST
000009               3,ISW1,ISW2,IGRDTP                                         MODR
000010                INTEGER ZERO                                              CONST
000011                COMMON /FLAGS/                                            FLAGS
000012               1IERR,IPRNT,GOOD1,NSHC,NSSC,NSEGC,NSUBR,NSEGR,NRSIDF,NRCWF,NRSHF, FLAGS
000013               2ISSF,ISTAT,IREGL,IZONFL,ISTRL,ISUBL,ISEGL,NSEG,IMXSEG,NTLIST, FLAGS
000014               3READSF,GOOD2,FATAL,IDAYS(8),IDSEG(150)                    MOD2
000015               1,IM01(8),IDA1(8),IYR1(8),IYR1(8),IM01(8),IDA1(8)          FLAGS
000016                INTEGER GOOD1,GOOD2,READSF,FATAL                          INPUT
000017                COMMON /INPUT/                                            INPUT
000018               1RSEED,SUBAR,HTSTPS,DEVTPM,CV(5),DEVTPM,QLAT,QLONG,SLAT(5),SLONG(5) INPUT
000019               2,NAMSUB(2),ICTRY,ICTRY2,ICASL,HIST,ITSEG,ISHF,ISSD,ISID,ISCW, INPUT
000020               3ISEG,IREG,IZONE,ISTRAT,ISUB,ISW,ITYPE,ITYPE2,IMDEX,ISEG2,IREG2, INPUT
000021               4ZONE2,ISTRA2,ISUB2,LAT(3),LONG(3),IYR(8),IMO(8),IDA(8),CV4,ICRP, INPUT
000022               5DELTHF(8),IPLNTF,ISEGSD,ITRIND,NAGR,CAL,ITLIST(6),NCRCD,NCHECK, INPUT
000023               6NCROP,IDAYS1(8),DELTHF1(8),IPLNT1,ISGSD1,IDAS(8),IMOS(8),IYRS(8) INPUT
000024                DOUBLE PRECISION RSEED                                    INPUT
000025                REAL NAMSUB,ICTRY,ICTRY2                                  INPUT
000026                COMMON /CALEND/                                           CALEND
000027               1IPHDR(5),IPREDS(8),ICRPEW(5),ICRPES(5)                    CALEND
000028                COMMON /LDATA/                                            LDATA
000029               1WHDATA(8),SHDATA(8)                                       LDATA
000030                INTEGER WHDATA,SHDATA                                     LDATA
000031                REAL ICTRY1                                               LUMP
000032          C                                                               LUMP
000033                DIMENSION  SIDHDR(2), CWHDR(2), SHHDR(2)                  LUMP
000034                DATA  SIDHDR, CWHDR, SHHDR  / 4HSEGM,5HENT, 8HCROP,4HWIND LUMP
000035               1  ,4HSUB ,4HHIST  /,ICV1,IDEVTP,IHISIP/4HCV1 ,6HDEVTP4,6HHISIPH/ LUMP
000036                DATA IXPAR,ISIGMA/4HXPAR,5HSIGMA/                         LUMP
000037          C                                                               LUMP
000038          C                                                               LUMP
000039                REWIND ISIDF                                              LUMP
000040                REWIND ICWF                                               LUMP
000041                REWIND ISHF                                               LUMP
000042          C        WRITE HEADER RECORDS                                   LUMP
000043                NFILL= LRSIDF - 3                                         LUMP
000044                WRITE(ISIDF)SIDHDR,ICASE,ITSEG,(ZERO,I=1,13)              MOD5
000045                NFILL= LRCWF - 4                                          LUMP
000046                WRITE (ICWF)  CWHDR,ICASE,IWIN, ( ZERO,I=1,NFILL )        LUMP
000047                LRSHF = MXSEG +18                                         MOD5
000048                NFILL= LRSHF - 4                                          LUMP
000049                WRITE (ISHF)  SHHDR,ICASE,MXSEG, ( ZERO,I=1,NFILL )       LUMP
000050          C        INITIALIZE COUNTERS                                    LUMP
000051                ICTRY2 = 0                                                LUMP
000052                READSF = 0                                                LUMP
000053                NSEG= MXSEG                                               LUMP
000054                ICRPR = 0                                                 LUMP
000055                ICRPPR = 0                                                LUMP
000056                IREGP = 0                                                 LUMP
000057                IZONP = 0                                                 LUMP
000058                ISTRP = 0                                                 LUMP
```

```
000059          IGRP1 = 0                                                       MODS
000060    C                                                                     LUMP
000061    C        LOOP TO READ THE SUBSTRATA INTERMEDIATE FILE                 LUMP
000062     200  READ (INTSUB)  ICTRY,IREG,IZONE,ISTRAT,ISUB,NAMSUB,AREA,HISTPW  LUMP
000063        1  ,DEVTPW,CV,DEVTPM,NAGP,NAL,CV4,IGRP                            LUMP
000064    C                                                                     LUMP
000065    C        TEST FOR END OF DATA ON SUBSTRATA INTERMEDIATE FILE          LUMP
000066     203  CONTINUE                                                        LUMP
000067    C                                                                     LUMP
000068          IF ( ICTRY .NE. ENDATA )  GO TO 205                             LUMP
000069          ICRPPR = 1                                                      LUMP
000070     204  IF ( ILIST .NE. 0 )  WRITE (IOUT,11)                            LUMP
000071     11   FORMAT (//44H  END OF DATA ON SUBSTRATA INTERMEDIATE FILE)      LUMP
000072    C                                                                     LUMP
000073    C        IF EOF ALREADY DETECTED ON SEGMENT INTERMEDIATE FILE,        LUMP
000074    C        THEN TERMINATE OUTPUT FILES AND EXIT.                        LUMP
000075    C        OTHERWISE CONTINUE READING REMAINDER OF SEGMENT INTERMEDIATE LUMP
000076    C        FILE.                                                        LUMP
000077          IF ( ICTRY2 .EQ. ENDATA )  GO TO 800                            LUMP
000078          IF(IGRP1 .EQ. 0)GO TO 100                                       MODS
000079          IF(IGRP2 .NE. 0)GO TO 100                                       MODS
000080          WRITE(IOUT,16)IREGP,IZONP,ISTRP                                 MODS
000081          WRITE(IOUT,19)                                                  MODS
000082      100 GO TO 290                                                       MODS
000083    C                                                                     LUMP
000084      205 IF(ICRPR .NE. 0)GO TO 206                                       MODS
000085          READ(ICALTH)ICTRY1,IREG1,IZON1,ISTRA1,ISUB1,(IDAYS(I),I=1,8)    LUMP
000086         1,(DELTHF(I),I=1,4),IPLNTL,ISEGSD,(IDAYS1(I),I=1,8),             LUMP
000087         2(DELTHF1(I),I=1,4),IPLNT1,ISEGS01                               LUMP
000088      206 IF(ICTRY1 .EQ. ENDATA)GO TO 211                                 MODS
000089          IF(IREG - IREG1)211,207,212                                     MODS
000090      207 IF(IZONE - IZON1)211,208,212                                    LUMP
000091      208 IF(ISTRAT - ISTRA1)211,209,212                                  LUMP
000092      209 IF(ISUB - ISUB1)211,230,212                                     LUMP
000093      230 ICRPR = 0                                                       LUMP
000094          ICRPPR = 0                                                      LUMP
000095          GO TO 215                                                       LUMP
000096      212 ICRPR = 0                                                       MODS
000097          GO TO 213                                                       LUMP
000098      211 ICRPR = 1                                                       MODS
000099      213 ICRPPR = 1                                                      LUMP
000100      215 ISSUB = 0                                                       LUMP
000101          IF(IREG .NE. IREGP)GO TO 235                                    LUMP
000102          IF(IZONE .NE. IZONP)GO TO 235                                   LUMP
000103          IF(ISTRAT .EQ. ISTRP)GO TO 240                                  LUMP
000104      235 IF(IGRP1 .EQ. 0)GO TO 237                                       MODS
000105          IF(IGRP2 .NE. 0)GO TO 237                                       MODS
000106          WRITE(IOUT,16)IREGP,IZONP,ISTRP                                 MODS
000107          WRITE(IOUT,19)                                                  MODS
000108      237 IGRP1 = 0                                                       MODS
000109          IREGP = IREG                                                    MODS
000110          IZONP = IZONE                                                   LUMP
000111          ISTRP = ISTRAT                                                  LUMP
000112          IGRP2 = 0                                                       LUMP
000113      240 IF(IGRP .EQ. 2)IGRP1 = 1                                        MODS
000114          IF(ICRPPR .NE. 0)GO TO 242                                      MODS
000115          DO 241 I=1,8                                                    MODS
000116          IPREDS(I) = 0                                                   MODS
000117          IPREDS(I) = 0                                                   MODS
000118      241 CONTINUE                                                        MODS
```

```
      DO 243 I=1,5                                                      *NEW
      ICRPEW(I) = 0                                                     *NEW
      ICRPES(I) = 0                                                     *NEW
  243 CONTINUE                                                          *NEW
      CALL CALERR(IDAYS,DELTWE,IPLNTE,IPREDW,ICRPEW)            LUMP    **-2
      CALL CALERR(IDAYS1,DELWE1,IPLNT1,IPREDS,ICRPES)           LUMP
  242 XBAR =HISTPW*(1.0 + DEVTPW)/100.0                         LUMP
      SIGMA = HISTPW*CV(1)/100.0                                MOD1
      ICPT = 0                                                  LUMP
      IFR=0                                                     LUMP
      CALL BETAD(RSEED,XBAR,SIGMA,XI,IOPT,IFR)                  LUMP
      IF(IER.EQ.0) GO TO 260                                    LUMP
      IF(IER.EQ.1) GO TO 245                                    LUMP
      IF(IER.EQ.2) GO TO 250                                    LUMP
      WRITE(IOUT,900) IREG,IZONE,ISTRAT,ISUB                    LUMP
  900 FORMAT(1H0,8H REGION,I3,7H  ZONE ,I3,8H STRATA ,I4,6H SUBS ,I4,  LUMP
     1/,48H  BETA FUNCTION COULD NOT CONVERGE ON A SOLUTION)    MOD3
      TPWSS=HISTPW                                              LUMP
      GO TO 265                                                 LUMP
  245 IPRNT2=IDEVTP                                             LUMP
      PRNT4 = DEVTPW                                            MOD3
      IPRNT3=IHISTP                                             LUMP
      PRNT5=HISTPW                                              LUMP
      IPRNT1=IXBAR                                              LUMP
      GO TO 255                                                 LUMP
  250 IPRNT1 = ISIGMA                                           MOD1
      IPRNT2=ICV1                                               LUMP
      IPRNT3=IHISTP                                             LUMP
      PRNT4 = CV(1)                                             MOD1
      PRNT5=HISTPW                                              LUMP
  255 WRITE(IOUT,905)IREG,IZONE,ISTRAT,ISUB,IPRNT1,IPRNT2,IPRNT3,  LUMP
     1PRNT4,PRNT5                                               LUMP
  905 FORMAT(1H0,8H REGION,I3,6H ZONE ,I3,8H STRATA ,I4,6H SUBS ,I4,/,  LUMP
     122H  BETA FUNCTION RESET ,A6,33H VALUE SINCE NOT BETWEEN 0 AND 1.,LUMP
     2/,9H  EITHER ,A6,4H OR ,A6,17H HAVE BAD VALUES ,2F8.3)    MOD3
  260 TPWSS=XI*100.                                             LUMP
  265 IF ( NSEG .EQ. 0 )  GO TO 220                             LUMP
      DO 210 I=1,NSEG                                           LUMP
      IPSEG(I)= 0                                               LUMP
  210 CONTINUE                                                  LUMP
      NSEG= 0                                                   LUMP
C                                                               LUMP
C        IF EOF ALREADY DETECTED ON SEGMENT INTERMEDIATE FILE,  LUMP
C        THEN WRITE HISTORICAL FILE AND CONTINUE READING SUBSTRATA  LUMP
C        INTERMEDIATE FILE.                                     LUMP
  220 IF ( ICTRY2 .EQ. ENDATA )  GO TO 450                      LUMP
C        IF NEADSE .NE. 0, SKIP READING SEGMENT INTERMEDIATE FILE  LUMP
  290 IF ( NEADSE .NE. 0 ) GO TO 510                            LUMP
C                                                               LUMP
C        LOOP TO READ THE SEGMENT INTERMEDIATE FILE.            LUMP
  300 ISEGR= ISEGR + 1                                          LUMP
      READ (IUNSEG)  ICTRY2,ISEG,IREG2,IZONE2,ISTRA2,ISUB2,QLAT,QLONG  LUMP
     1  ,ISW,ITRIND,(ITLIST(I),I=1,6)                           LUMP
C                                                               LUMP
C        TEST FOR END OF DATA ON SEGMENT INTERMEDIATE FILE.     LUMP
  305 CONTINUE                                                  LUMP    **-1
C                                                               LUMP
      IF ( ICTRY2 .NE. ENDATA )  GO TO 305                      LUMP
  308 IF ( TLIST .NE. 0 )  WRITE (IOUT,12)                      LUMP
   12 FORMAT (//42H  END OF DATA ON SEGMENT INTERMEDIATE FILE)  LUMP
```

```
000179  C        IF EOF ALREADY DETECTED ON SUBSTRATA INTEMEDIATE FILE,          LUMP
000180  C        THEN TERMINATE OUTPUT FILES AND EXIT.                           LUMP
000181  C        OTHERWISE WRITE ON CROP WINDOW FILE AND HISTORICAL FILE.        LUMP
000182        IF ( ICTRY .EQ. ENDATA )  GO TO 800                                LUMP
000183        GO TO 430                                                          LUMP
000184  C                                                                        LUMP
000185   305  READSF= 1                                                          LUMP
000186  C        IF EOF ALREADY DETECTED ON SUBSTRATA INTERMEDIATE FILE,         LUMP
000187  C        THEN READ REMAINDER OF SEGMENT INTERMEDIATE FILE WITHOUT        LUMP
000188  C        WRITING ON OUTPUT FILES.                                        LUMP
000189   310  IF ( ICTRY .EQ. ENDATA )  GO TO 410                                LUMP
000190  C                                                                        LUMP
000191  C        DOES THE CURRENT SEGMENT BELONG TO THE CURRENT SUBSTRATA RECORDLUMP
000192        IF ( IREG - IREG2 )  430,312,400                                   LUMP
000193   312  IF ( IZONE - IZONE2 )  430,314,400                                LUMP
000194   314  IF ( ISTRAT - ISTRA2 )  430,316,400                               LUMP
000195   316  IF ( ISUB - ISUB2 )  430,318,400                                  LUMP
000196  C        YES.  NOW SEE IF THE CURRENT SEGMENT IS IN THE TRAINING         LUMP
000197  C        INDICATOR LIST.                                                 LUMP
000198   318 IF(TCRPPR.NE.0) GO TO 415                                           LUMP
000199        IF(ISW.EQ.1HS)ISWH=1                                               LUMP
000200        IF(ISW.EQ.1HW)ISWH=0                                               LUMP
000201        IF(ISWH.EQ.1.AND.IDAYS1(I).EQ.0) GO TO 415                         LUMP
000202        IF(ISWH.EQ.0.AND.IDAYS(I).EQ.0) GO TO 415                          LUMP
000203        ISSUB=ISSUB+1                                                      LUMP
000204   300  NSEG= NSEG + 1                                                    LUMP
000205         IDSEG(NSEG)= ISEG                                                LUMP
000206        CALL GRID(IERT)                                                    MODB
000207        IF(IERT .NE. 0)GO TO 300                                           MODB
000208  C                                                                        LUMP
000209  C        WRITE CURRENT SEGMENT ON SEGMENT ID FILE                        LUMP
000210        WRITE(ISIMF)ICTRY,IREG,IZONE,ISTRAT,ISUB,ISEG,ITRIND,(ITLIST(I),   MOD3
000211       1I=1,6),BLAT,BLONG,INDEX,ISWH                                       MOD3
000212  C****                                                                    LUMP
000213        IF ( MTST .EQ. 2 )                                                 LUMP
000214       1WRITE(IOUT,21)ICTRY,IREG,IZONE,ISTRAT,ISUB,ISEG,ITRIND,(ITLIST(I) MOD3
000215       2,I=1,6),BLAT,BLONG,INDEX,ISWH                                      MOD3
000216    21 FORMAT(1H0,17H  SEGMENT ID FILE/,                                   *NEW
000217       19H  ICTRY =,A8,9H   IREG =,I4,10H   IZONE =,I4,                    *NEW
000218       21H   ISTRAT =,                                                     *NEW
000219       115,9H   ISUB =,I5,9H   ISEG =,I5,11H   ITRIND =,I2/                MOD3  **-2
000220       211H   ITLIST =,6I5/9H   BLAT =,F14.6,10H   BLONG =,E14.6,          MOD3
000221       310H   INDEX =,I5,9H   ISWH =,I2)                                   MOD3
000222        NRSIDF= NRSIDF + 1                                                 LUMP
000223        READSF= 0.                                                         LUMP
000224        GO TO 300                                                          LUMP
000225  C                                                                        LUMP
000226  C        INCONSISTENCY BETWEEN SUBSTRATA DATA AND SEGMENT DATA.          LUMP
000227   400  WRITE (IOUT,13)  ISEG,ICTRY2,IREG2,IZONE2,ISTRA2,ISUB2             LUMP
000228       1  ,ICTRY,IREG,IZONE,ISTRAT,ISUB                                    LUMP
000229    13  FORMAT(//53H THE FOLLOWING SEGMENT DATA WAS DROPPED DUE TO DISAGR LUMP
000230       1 31MEMENT WITH THE SUBSTRATA DATA./10H   SEGMENT,I5,11H   COUNTRYLUMP
000231       2 ,A8,9H   REGION,I4,7H   ZONE,I4,10H   STRATUM,I5,13H   SUBSTRATUMLUMP
000232       3 ,I5,19H  FROM SEGMENT DATA/15X,11H   COUNTRY ,A8,9H   REGION,I4  LUMP
000233       4 ,7H   ZONE,I4,10H   STRATUM,I5,13H   SUBSTRATUM,I5,21H  FROM SUBSLUMP
000234       51RATA DATA)                                                        LUMP
000235        GO TO 300                                                          LUMP
000236  C                                                                        LUMP
000237   410  WRITE (IOUT,14)  ISEG,ICTRY2,IREG2,IZONE2,ISTRA2,ISUB2            LUMP
000238    14  FORMAT (//51H THE FOLLOWING SEGMENT DATA WAS DROPPED BECAUSE IT    LUMP
```

```
000239          1  6HNOTD NOT BELONG TO ANY STRATA ON THE SUBSTRATA HISTORICAL FILE LUMP
000240          2  /10H   SEGMENT,I5,11H   COUNTRY ,A4,9H   REGION,I4,7H   ZONE,I4 LUMP
000241          3  ,10H   STRATUM,I5,13H   SUBSTRATUM,I5)                          LUMP
000242           GO TO 300                                                         LUMP
000243      415 WRITE(IOUT,15)IREG,IZONE,ISTRAT,ISUB,ISEG                          LUMP
000244       15 FORMAT(1H0,8H  REGION ,I3,6H ZONE ,I3,8H STRATA ,I4,11H SUBSTRATA  LUMP
000245         1,I4,9H SEGMENT ,I4,/,93H NO CROP CALENDER DATA AVAILABLE FOR SEGME MOD4
000246         2NT INTERMEDIATE FILE RECORD. SEGMENT RECORD SKIPPED )              MOD3
000247           GO TO 300                                                         LUMP
000248  C                                                                          LUMP
000249  C          WRITE RECORD ON CROP WINDOW FILE                                LUMP
000250      430 CONTINUE                                                           MOD3
000251           IF(IGRP .EQ. 1)GO TO 440                                          LUMP
000252           IF(IGRP.EQ.2) GO TO 445                                           LUMP
000253           IF(ISSUB.EQ.0)GO TO 448                                           LUMP
000254           FATAL=1                                                           LUMP
000255           WRITE(IOUT,16)IREG,IZONE,ISTRAT,ISUB                              LUMP
000256       16 FORMAT(1H0,8H REGION ,I3,6H ZONE ,I3,8H STRATA ,I4,11H SUBSTRATA  LUMP
000257         1,I4)                                                               LUMP
000258           WRITE(IOUT,17)                                                    LUMP
000259       17 FORMAT(51H  A GROUP 3 SUBSTRATA HAS 1 OR MORE SAMPLE SEGMENTS)     LUMP
000260           GO TO 448                                                         LUMP
000261      440 IF(ISSUB.NE.0) GO TO 448                                           LUMP
000262           IGRP=3                                                            LUMP
000263           WRITE(IOUT,16)IREG,IZONE,ISTRAT,ISUB                              LUMP
000264           WRITE(IOUT,18)                                                    LUMP
000265       18 FORMAT(61H  A GROUP 1 SUBSTRATA DOES NOT HAVE AT LEAST 1 SAMPLE SE LUMP
000266         1GMENT)                                                             LUMP
000267           GO TO 448                                                         LUMP
000268      445 IF(IREG.NE.IREG2) GO TO 447                                        LUMP
000269           IF(IZONE .NE. IZONE2) GO TO 447                                   LUMP
000270           IF(ISTRAT .NE. ISTRA2) GO TO 447                                  LUMP
000271           IGRP2= IGRP2+ ISSUB                                               LUMP
000272           GO TO 448                                                         LUMP
000273      447 CONTINUE                                                           MOD5
000274       19 FORMAT(49H  A STRATA WITH AT LEAST 1 GROUP 2 SUBSTRATA DOES,       MOD1
000275         1 5H NOT HAVE A SAMPLE SEGMENT IN ANY GROUP 2 SUBSTRATA)            MOD1
000276      448 IF(ICRPER .NE. 0)GO TO 450                                         MOD3
000277           NRCWF = NRCWF + 1                                                 MOD3
000278           WRITE (ICWF)  ICTRY,IREG,IZONE,ISTRAT,ISUB, (IPREDW(I),I=1,8),    LUMP
000279         1ISEGSD,(ICRPEW(I),I=1,5),(IPREDS(I),I=1,8),ISGSDI,(ICRPES(I),I=1,5LUMP
000280         2)                                                                  LUMP
000281  C*****                                                                     LUMP
000282           IF ( ILIST .EQ. 2 )                                               LUMP
000283          1WRITE(IOUT,24)ICTRY,IREG,IZONE,ISTRAT,ISUB,IPREDW,ISEGSD,         MOD3
000284          2ICRPEW,IPREDS,ISGSDI,ICRPES                                       LUMP
000285       24 FORMAT(1H0,10H  CROP WINDOW FILE/10H   ICTRY =,A4,9H   IREG =,I4,  *NEW
000286          110H   IZONE =,I4,11H  ISTRAT =,I5,9H   ISUB =,I5/                 *NEW
000287          210H IPREDW = ,8I6/10H ISEGSD = ,I2,10H ICRPEW = ,5I3/,            LUMP **-2
000288          310H IPREDS = ,8I6/,10H ISGSDI = ,I2,10H ICRPES = ,5I3/)           LUMP
000289  C          WRITE RECORD ON SUBSTRATA HISTORICAL FILE                       LUMP
000290      450 NRSHF = NRSHF + 1                                                  LUMP
000291           WRITE (ISHF)  ICTRY,IREG,IZONE,ISTRAT,ISUB,NSEG,( IDSEG(I),I=1,   LUMP
000292         1  NSEG ),IGRP,   HISTPW,AREA,IPHSS,NAGR,NAI,DEVTPW,DEVTPM,CV,CV4 LUMP
000293  C*****                                                                     LUMP
000294           IF ( ILIST .NE. 2 )  GO TO 200                                    LUMP
000295           WRITE(IOUT,22)ICTRY,IREG,IZONE,ISTRAT,ISUB,NSEG,(IDSEG(I),I=1,    MOD3
000296         1       NSEG )                                                      LUMP
000297       22 FORMAT(1H0,27H  SUBSTRATA HISTORICAL FILE/10H   ICTRY =,A4,        *NEW
000298          19H   IREG =,I4,10H   IZONE =,I4,11H   ISTRAT =,I5,9H   ISUB =,I5, *NEW
```

```
      29H   NSEG =,I4/10H   IDSEG =,16I6/(10X,16I6))                          *NEW
       WRITE(IOUT,23)IGRP,HISTPW,AREA,TPWSS,NAGR,NAL,DEVTPW,DEVTPM,      MOD3   **-3
      1CV,CV4                                                            LUMP
   23 FORMAT(11H    IGRP  =,I2,11H   HISTPW =,E14.6,9H   AREA =,E14.6,        *NEW
      110H   TPWSS =,E14.6/10H NAGR   = ,I5,10H NAL     = ,I3,              *NEW
      110H DEVTPW = ,F6.3,10H DEVTPM = ,F6.3,/10H CV     = ,4(F5.3,1X))  MOD4   **-2
       GO TO 200                                                         LUMP
C                                                                        LUMP
C        WRITE TERMINAL RECORDS ON OUTPUT FILES.                         LUMP
  800  NFILL= LRSIDF - 1                                                 LUMP
       WRITE (ISIDF)  ENDATA, ( ZERO,I=1,NFILL )                         LUMP
       REWIND ISIDF                                                      LUMP
       NFILL= LRCWF - 1                                                  LUMP
       WRITE ( ICWF)  ENDATA, ( ZERO,I=1,NFILL )                         LUMP
       REWIND ICWF                                                       LUMP
       NFILL= LRSHF - 1                                                  LUMP
       WRITE (ISHF)  ENDATA, ( ZERO,I=1,NFILL )                          LUMP
       REWIND ISHF                                                       LUMP
       RETURN                                                            LUMP
       END                                                               LUMP
```

```
• FLT LUMPDR•1•760405• 55992   • 1
000001          C      PROGRAM LUMPDR(ITEST=401,OTEST=401,TAPE5=ITEST,TAPE6=OTEST          MOD1
000002          C     1,SHIST=401,SSTATD=401,SEGIDD=401,SEGCWD=401,SEGID=604,CHOPW=604,    MODR
000003          C     2SUBHST=604,CRPERR=401,CRPINT=604,TAPE30=504,TAPE31=504,             MODR
000004          C     3INDMAT=1004,TAPE4=INDMAT                                            MODR
000005          C     2  ,TAPE11=SHISTD,TAPE12=SSTATD,TAPE13=SEGIDD,TAPE14=SEGCWD          LUMPDR
000006          C     4,TAPE1=SEGID,TAPE2=CHOPW,TAPE3=SUBHST,                              MODR
000007          C     4TAPE36=CRPERR,TAPE37=CRPINT)                                        LUMPDR
000008                COMMON /CONST/                                                       CONST
000009               1PI,DEGTOR,       SLATMX,ENDATA,ZERO,INP,IOUT,INTSUB,INTSEG,ISIDF,    CONST
000010               2ICWF,ICHF,LRSIDF,LRCWF,LRSHF,IWIN,MXSEG,ICALIN,ICWE,AREACF          CONST
000011               3,ISN1,ISN2,IGRDIP                                                   MODR
000012                INTEGER ZERO                                                        CONST
000013                COMMON /FLAGS/                                                      FLAGS
000014               1IERR,IPRNT,GOOD1,NSHC,NSSC,NSFGC,NSUBR,NSFGR,NRSIDF,NRCWF,NRSHF,    FLAGS
000015               2ISSC,ISTAT,IREGI,IZONEL,ISTRL,ISUBL,ISEGL,NSEG,IMXSEG,NTLIST,       FLAGS
000016               3READSF,GOOD2,FATAL,IDAYS(8),IDSF,(150)                              FLAGS
000017               1,IMO1(8),IDA1(8),IYR1(8),IYR1(8),IMO1(8),IDA1(8)                    MOD2
000018                INTEGER GOOD1,GOOD2,READSF,FATAL                                    FLAGS
000019                COMMON /INPUT/                                                      INPUT
000020               1RSEED,SUBAR,HISTPW,DEVTPW,CV(3),DEVTPW,OLAT,OLONG,SLAT(3),SLONG(3)INPUT
000021               2,NARSUB(2),ICTRY,ICTRY2,ICASE,ILIST,ITSEG,ISHD,ISSD,ISID,ISCW,      INPUT
000022               3ISEG,IPRG,IZONE,ISTRAT,ISUB,ISN,ITYPE,ITYPE2,INDEX,ISEG2,IREG2,     INPUT
000023               4IZONE2,ISTRA2,ISUB2,LAT(3),LONG(3),IYR(8),IMO(8),IDA(8),CV4,IGRP,   INPUT
000024               5DELTWE(4),IPLNTF,ISEGSD,ITRIND,NAGR,NAL,ITLIST(6),NCHCD,NCHFCD,     INPUT
000025               6NCHOP,IDAYS1(8),DELWE1(4),IPLNT1,ISNSD1,IDAS(8),IMOS(8),IYRS(8)     INPUT
000026                DOUBLE PRECISION RSEED                                              INPUT
000027                REAL NARSUB,ICTRY,ICTRY2                                            INPUT
000028                COMMON /LDATA/                                                      LDATA
000029               1WWDATA(8),SWDATA(8)                                                 LDATA
000030                INTEGER WWDATA,SWDATA                                               LDATA
000031          C                                                                         LUMPDR
000032                DATA  PI          , DEGTOR       , AREACF      , SLATMX , ENDATA    LUMPDR
000033               1    / 3.1415926 , 0.017453292 , 4.046873E-3, 65.0    , 4HZZZZ  /  LUMPDR
000034                DATA  ZERO  , INP , IOUT , INTSUB , INTSEG , ISIDF , ICWF , ISHF    LUMPDR
000035               1    /   0   ,  5  ,   6  ,   13  ,   14  ,   1  ,   2  ,  3  /       LUMPDR
000036                DATA  LRSIDF , LRCWF , LRSHF , IWIN , MXSEG , ICWF , ICALIN         LUMPDR
000037               1    /   17   ,   35  ,  170  ,   4  ,  150  ,   8  ,   7  /          MOD1
000038          C                                                                         LUMPDR
000039                DATA FATAL/0/,IGRDIP/4/                                             MOD9
000040          C                                                                         LUMPDR
000041          C                                                                         LUMPDR
000042          C*****       TEMPORARILY REWIND INPUT FILES *****                         LUMPDR
000043          C*****                                                                    LUMPDR
000044          C******                                                                   LUMPDR
000045                REWIND INTSUB                                                       LUMPDR
000046                REWIND INTSEG                                                       LUMPDR
000047                REWIND ICALIN                                                       LUMPDR
000048          C                                                                         LUMPDR
000049          C        READ CONTROL CARD                                                LUMPDR
000050                READ (INP,1)  ICASE,RSEED,ILIST,ITSEG,ISHD,ISSD,ISID,ISCW,ICWE      LUMPDR
000051          1     FORMAT (I5,3X,D12.0,I7,6I5)                                         LUMPDR
000052                WRITE (IOUT,10)  ICASE,RSEED,ILIST,ITSEG,ISHD,ISSD,ISID,ISCW,ICWE   LUMPDR
000053         10     FORMAT (1H1  CASE NO.,T34                                           LUMPDR
000054               1  /2H0  INITIAL RANDOM NUMBER SEED,D19.12                            LUMPDR
000055               2  /1H0  LIST OPTION,T31                                              LUMPDR
000056               3  /2H0  TRAINING SEGMENT OPTION,I19                                  LUMPDR
000057               4  /4H0  SUBSTRATA HISTORICAL DATA FILE NO.,I8                        LUMPDR
000058               5  /3H0  SUBSTRATA STATISTICAL DATA FILE NO.,I7                       LUMPDR
```

```
      6  /34H0  SAMPLE SEGMENT ID DATA FILE NO.,I11                    LUMPDR
      7  /35H0  SUBSTRATA CROP CALENDER FILE NO.,5X,I5                 MOD2
      8  /36H0  CROP CALENDER ERROR DATA FILE NO.,4X,I5)               MOD2
C                                                                      LUMPDR
      REWIND TSHD                                                      LUMPDR
      REWIND ISSD                                                      LUMPDR
      REWIND ISTD                                                      LUMPDR
      REWIND ISCW                                                      LUMPDR
      REWIND ICWE                                                      LUMPDR
      CALL RDGRID(0)                                                   MODA
C                                                                      LUMPDR
      IF ( ILIST .NE. 0 )  WRITE (IOUT,11)  ICASE,RSEED,ILIST,ITSFG,ISHD LUMPDR
     1  ,ISSD,ISID,ISCW,ICWE                                           LUMPDR
   11 FORMAT (30H1 LISTING OF INPUT CARD IMAGES///16H    CONTROL CARD// LUMPDR
     11X,I5,3X,D19.12,I7,6I5)                                                *NEW
C                                                                      LUMPDR  **-1
   50 IF ( ISHD .NE. 0 .OR. ISID .NE. 0 )  CALL FILGEN                 LUMPDR
      CALL LUMP                                                        LUMPDR
      WRITE(IOUT,905)NRSIDF,NRCWF,NRSHF                                MOD3
  905 FORMAT(1H0,I4,32H SEGMENT ID FILE RECORDS WRITTEN/               MOD1
     1,1X,I4,35H CROP CALENDER FILE RECORDS WRITTEN/                   MOD1
     2,1X,I4,39H SEGMENT REFERENCE FILE RECORDS WRITTEN)               MOD1-
      IF(FATAL .NE. 0)WRITE(IOUT,900)                                  LUMPDR
  900 FORMAT(1H0,55H FATAL ERROR OCCURED IN LUMP, FILES MAY NOT BE ANY LUMPDR
     1OUD)                                                             LUMPDR
      WRITE (IOUT,13)  RSEED                                           LUMPDR
   13 FORMAT(1/H1  JOB COMPLETED.//28H   LAST RANDOM NUMBER SEED =     LUMPDR
     1D19.12)                                                          LUMPDR
      CALL RDGRID(-1)                                                  MODA
      STOP                                                             LUMPDR
      END                                                              LUMPDR
```

```
* ELT RDGRID,1,760401, 67419    , 1

      SUBROUTINE RDGRID(JG)                                             RDGRID
C     THIS ROUTINE WILL OPEN OR CLOSE IGRDTP FILE OR WRITE A RECORD ON IRDGRID
C     T                                                                 RDGRID
      COMMON /CONST/                                                    CONST
     1PI,DEGTOR,      SLATMX,FNDATA,ZERO,INP,IOUT,INTSUB,INTSEG,ISIDF,  CONST
     2ICWF,ISMF,LRSIDF,LRCWF,LRSHF,IWIN,MXSEG,ICALIN,ICWL,ARLAUF        CONST
     3,ISW1,ISW2,IGRDTP                                                 RDGR
      INTEGER ZERO                                                      CONST
      COMMON /DTRAC/                                                    DTRAC
     1IFDTN(252),IREC(500)                                              DTRAC
      DATA MAXJ/42/,IMAT1/4HINDE/,IMAT2/4HXMAT/                         RDGRID
C     DATA MAXJ/252/                                                    RDGRID
      IF(JG)25,10,40                                                    RDGRID
   10 DEFINE FILE 4(42,500,U,IDUM)
      READ(IGRDTP'1,ERR=100)(IREC(I),I=1,500)
      IF(IREC(1) .EQ. IMAT1 .AND. IREC(2) .EQ. IMAT2)GO TO 50           RDGRID
      WRITE(IOUT,900)                                                   RDGRID
      STOP                                                              RDGRID
  900 FORMAT(1H0,85H *** WRONG FILE MOUNTED FOR INDEX GRID MATRIX)      RDGRID
   25 CONTINUE
C  25 CALL CLOSE(IGRDTP,0)
      GO TO 50                                                          RDGRID
   40 J = JG + 1                                                        RDGRID
      READ(IGRDTP'J,ERR=100)(IREC(I),I=1,500)
   50 RETURN                                                            RDGRID
  100 WRITE(IOUT,901)
      STOP
  901 FORMAT(1H0,91H *** AN IRRECOVERABLE I/O ERROR HAS OCCURRED ON READ
     1ING A RECORD FROM THE INDEX MATRIX FILE /10X,31H THE JOB IS BEING
     2ABANDONED ***)
      END                                                               RDGRID
```

```
@ FLT RDM1A,1,760401, 67411     , 1

000001              SUBROUTINE RDM1A(FL,U)                                  RDM1A
000002        C     RANDOM NUMBER GENERATOR FOR ALL COMPUTERS               RDM1A
000003              DOUBLE PRECISION C1,C2,R1,R2,T,FL,TWO35,ONE,ZERO,XMOD,YMOD  RDM1A
000004              DATA C1,C2,ONE,ZERO/3051749376.D1,84365.D0,1.D0,0.D0/,  RDM1A
000005             1     TWO35/34359738368.D0/                              RDM1A
000006              XMOD(T) = DMOD(T,TWO35)                                 RDM1A
000007              YMOD(T) = DMOD(T,262144.D0)                             RDM1A
000008              T = FL                                                  RDM1A
000009              IF(T .LE. ZERO) T = ONE                                 RDM1A
000010              R2 = YMOD(T)                                            RDM1A
000011              R1 = T - R2                                             RDM1A
000012              T = XMOD(XMOD(C1*R2+C2*R1) + C2*R2)                     RDM1A
000013              U = T/TWO35                                             RDM1A
000014              FL = T                                                  RDM1A
000015              RETURN                                                  RDM1A
000016              END                                                     RDM1A

  11. TOC
```

SEE BOOK IV

Table of Contents

PART I

PROBLEM DESCRIPTION

Problem Description for System Error Executive (SEE) Program

1.0 SCOPE

1.1 PROGRAM CAPABILITIES

This program provides the means to create certain files containing error coefficients and terms necessary as input to the LEM program. The input data to this program consists of three card sets containing YES error and truth data, CAMS error data and signature extension data. As the data on the cards is transferred from card images to the output data files, each data field is checked for validity and each card ID group is checked against the substrata historical file to detect missing data. A message is printed for each error found. The program provides on option a list of all cards and also a listing of all records written on the output files. The output files are the YES error file, CAMS error file and signature extension file.

1.2 PROGRAM DEVELOPMENT AND ORGANIZATION

The program will be developed in FORTRAN IV on the CDC 6600 time sharing system and later converted to FORTRAN V in order to be run on the UNIVAC 1110 at NASA, Houston, under EXEC VIII.

1.3 OPERATIONAL ASSUMPTIONS

- One case may be run at one time.
- Only one country may be processed in a case.
- The three data card sets must be on separate input files and in sort as follows:
  The major sort field is listed first.

  1. YES error data, set 5

     Country, region, zone, strata and card sequence number

  2. CAMS error data set 6

     Country, region, zone, strata, substrata, segment and card sequence number

3. Signature extension set 7

Country, region, zone and card sequence number

- All cards in a group with the same ID must be present even though a card may contain only blank data entries.
- The program will require less than 20,000 words decimal of storage in the CPU of the UNIVAC.
- The program will run only if the specified substrata historical file is present.
- Except in the case of an input card set being out of sort, the program will continue checking for field errors and for mismatches vs. the substrata historical file. A fatal error will stop output files from being written.
- Input data groups for which there is no ID match on the substrata historical file are considered to be extra data and are not written out on the output files.
- IDFRS and IDTOS specify ID limits for which certain errors are considered to be fatal. If a normally fatal error occurs on a card group outside these limits, it will be considered to be non-fatal.
- A negative non-zero entry in any card input unit designator IYESR, ISIGEX or ICAMER indicates that that data is not to be read or processed and the corresponding output file will not be created.
- The United States must be assigned the symbol "USA " and Canada must be assigned the symbol "CAND".

2.0 INPUT

There are four card sets and one binary data file required as input to this program.

The card sets are:

1. Program control card
2. YES truth and error data
3. CAMS error data
4. Signature extension data

The data file is:

1. Substrata historical file

2.1 CARD INPUT

2.1.1 List of Data Quantities

See input data description sheets on Pages 259-263.

2.1.2 Card Formats

See card format sheets on Pages 264-267.

2.1.3 Deck Setup

The input card data stream will have one card, the program control card.

The YES data will be on the file specified in IYESR (Unit 1 is assumed if this entry is 0).

The CAMS error data will be on the file specified in ICAMER (Unit 4 is assumed if this entry is 0).

The signature extension data will be on the file specified in ISIGEX (Unit 2 is assumed if this entry is 0).

In the case of the signature extension and CAMS error data cards, if the data is constant for a particular ID level such as zone, only one card group may be filled out for that zone and all ID levels below zone are left blank.

INPUT DATA DESCRIPTION

Name	Symbol	Dimension	Nominal Value	Range	Units	Description
CASE	-	1	-	1-9999	-	Case no. to be assigned to identify the three output files.
LIST	-	1	-	0-2	-	List option to list all input data cards or only those in error and all output file records. = 0 list cards in error = 1 list all input data cards = 2 list all input data cards and all output file records
DFRS	-	2	0, 0		-	Specifies ID of starting zone for which errors are to be considered fatal. 1st item region, 2nd - zone. Blank or 0 entry means entire ID range is active.
DTOS	-	2	0, 0		-	Specifies ID of ending zone for which errors are considered to be fatal. 1st item is reg., 2nd - zone.
YESU	-	1	1	-	-	File unit no. for YES input card data. Minus value means not to read or process this data.
SIGEX	-	1	2	-	-	File unit no. for signature extension input card data. Minus value means not to read or process this data.
CAMER	-	1	4	-	-	File unit no. for CAMS error input card data. Minus value means not to read or process.
CSESH	-	1	-	-	-	Zero or blank means not to check on case no. of input substrata historical file.
CTRY	-	1	-	-	-	Four character country name (i.e., USA , USSR).

DESCRIPTION

Units	Description
-	Region no.
-	Zone no.
-	Strata no.
-	Substrata no.
-	Segment no.
Quin/ hectar	True yield of stratum (for USA , and CAND - bushels/acre.
Yr, Mon, Day	Year, month and day for error truncation. If 1st value is 0, then there is no bias or std. dev. data. Any other 0 value terminates data entries.
Quin/ hectar	Bias error of yield. One value for each ≠0 value of IERDTE for USA , CAND - bushels/ acre.
Quin/ hectar	Standard deviation of yield error. One value for each ≠0 value of IERDTE for USA , CAND - bushels/acre.
-	Nominal frequency of labeling wheat given pure wheat pixels (conditional prob.). One for each window.
-	Nominal frequency of labeling wheat given mixed pixels (cond. prob.). One for each window.
-	Nominal frequency of labeling wheat given pure other pixels (cond. prob.). One for each window.

INPUT DATA DESCRIPTION

Name	Symbol	Dimension	Nominal Value	Range	Units	Description
BWW	$B(W/W)_i$	4	0	±9.999	-	Bias error in classifying pure wheat pixels. One for each window.
BWM	$B(W/M)_i$	4	0	±9.999	-	Bias error in classifying mixed pixels. One for each window.
BWO	$B(W/O)_i$	4	0	±9.999	-	Bias error in classifying other crop pixels. One for each window.
SIGWW	$\sigma(W/W)_i$	4	0	0-9.99	-	Standard deviation in classifying wheat pixels. One for each window.
SIGWM	$\sigma(W/M)_i$	4	0	0-9.99	-	Standard deviation in classifying mixed pixels. One for each window.
SIGWO	$\sigma(W/O)_i$	4	0	0-9.99	-	Standard deviation in classifying other crop pixels. One for each window.
BPW	B_{PW_i}	4	0	±9.999	-	Bias in proportion estimate (model #2). One for each window.
SIGPW	σ_{PW_i}	4	0	0-9.99	-	Standard deviation in proportion estimate (model #2).
			FOR MODEL 1 SIGNATURE EXTENSION			
B1W	B_{1W}	1	0	±9.999	-	Multiplicative bias error W/W.
B2W	B_{2W}	1	0	±9.999	-	Additive bias error W/W.
SIG1W	σ_{1iW}	6	-	0-9.99	-	Multiplicative std. dev. error (W/W). One for each training priority segment.
SIG2W	σ_{2iW}	6	-	0-9.99	-	Additive std. dev. error (W/W). One for each training priority segment.

INPUT DATA DESCRIPTION

Name	Symbol	Dimension	Nominal Value	Range	Units	Description
B1M	B_{1M}	1	-	±9.999	-	Multiplicative bias error (W/M).
B2M	B_{2M}	1	-	±9.999	-	Additive bias error (W/M).
SIG1M	σ_{1iM}	6	-	0-9.99	-	Multiplicative std. dev. error (W/M). One for each training priority segment.
SIG2M	σ_{2iM}	6	-	0-9.99	-	Additive std. dev. error (W/M). One for each training priority segment.
B1O	B_{1O}	1	-	±9.999	-	Multiplicative bias error (W/O).
B2O	B_{2O}	1	-	±9.999	-	Additive bias error (W/O).
SIG1O	σ_{1iO}	6	-	0-9.99	-	Multiplicative std. dev. error (W/O). One for each training priority segment.
SIG2O	σ_{2iO}	6	-	0-9.99	-	Additive std. dev. error (W/O). One for each training priority segment.
			FOR MODEL 2 SIGNATURE EXTENSION			
B1	B_1	1	-	±9.999	-	Multiplicative bias error
B2	B_2	1	-	±9.999	-	Additive bias error
SIG1	σ_{1i}	6	-	0-9.99	-	Multiplicative std. dev. error. One for each training priority segment.
SIG2	σ_{2i}	6	-	0-9.99	-	Additive std. dev. error. One for each training priority segment.
ISET	-	1	-	-	-	Card set number in column 80. 5 for YES error data. 6 for CAMS error data. 7 for signature extension data.

INPUT DATA DESCRIPTION

Name	Symbol	Dimension	Nominal Value	Range	Units	Description
SEQ	-	1	-	-	-	Card sequence number within card set with the same ID. Used to identify each card of a group. YES error has a 2 card group. CAMS error and sig. ext. each have 4 card groups.

SEE Control Card

1	5	6	9	12	15	18	20	22	24
		IDFRS		IDTOS					
ICASE	ILIST	Region	Zone	Region	Zone	IYESU	ISIGEX	ICAMER	ICSESH
I4	I1	I3	I3	I3	I3	I2	I2	I2	I4

YES Error Data Card 1

1	5	9	12	15	19	24	30	36	42	48	54	60	79	80
X	Country	Reg	Zone	Strata	Y	Error Trunc. Date 1	Date 2	Date 3	Date 4	Date 5	Date 6			
						Y M D	Y M D	Y M D	Y M D	Y M D	Y M D		1	5
	A4	I3	I3	I4	F5.2	3I2	3I2	3I2	3I2	3I2	3I2			

YES Error Data Card 2

1	5	9	12	15	19	24	29	34	39	44	49	54	59	64	69	74	79	80
X	Country	Reg	Zone	Strata	Bias Error of Yield						Standard Error of Yield							
					B_1	B_2	B_3	B_4	B_5	B_6	σ_1	σ_2	σ_3	σ_4	σ_5	σ_6	2	5
	A4	I3	I3	I4	6F5.1						6F5.2							

SEE Data Cards

CAMS Error Data Card 1 for Pure Wheat Pixels

5	9	12	15	19	23	27	31	35	39	45	51	57	63	67	71	75	79	80
					Nominal Freq. Label Wht.				Bias Error in Classifying Wheat				Std. Dev. Classify Wht.					
Country	Reg.	Zone	Strata	Sub-strata	$P(W/W)_1$	$P(W/W)_2$	$P(W/W)_3$	$P(W/W)_4$	$B(W/W)_1$	$B(W/W)_2$	$B(W/W)_3$	$B(W/W)_4$	$\sigma(W/W)_1$	$\sigma(W/W)_2$	$\sigma(W/W)_3$	$\sigma(W/W)_4$	1	6
A4	I3	I3	I4	I4	4F4.2				4F6.3				4F4.2					

CAMS Error Data Card 2 for Mixed Pixels

5	9	12	15	19	23	27	31	35	39	45	51	57	63	67	71	75	79	80
					Nominal Freq. Label Wht.				Bias Error in Classifying Wheat				Std. Dev. Classify Wht.					
Country	Reg.	Zone	Strata	Sub-strata	$P(W/M)_1$	$P(W/M)_2$	$P(W/M)_3$	$P(W/M)_4$	$B(W/M)_1$	$B(W/M)_2$	$B(W/M)_3$	$B(W/M)_4$	$\sigma(W/M)_1$	$\sigma(W/M)_2$	$\sigma(W/M)_3$	$\sigma(W/M)_4$	2	6
A4	I3	I3	I4	I4	4F4.2				4F6.3				4F4.2					

CAMS Error Data Card 3 for Other Pixels

5	9	12	15	19	23	27	31	35	39	45	51	57	63	67	71	75	79	80
					Nominal Freq. Label Wht.				Bias Error in Classifying Wheat				Std. Dev. Classify Wht.					
Country	Reg.	Zone	Strata	Sub-strata	$P(W/O)_1$	$P(W/O)_2$	$P(W/O)_3$	$P(W/O)_4$	$B(W/O)_1$	$B(W/O)_2$	$B(W/O)_3$	$B(W/O)_4$	$\sigma(W/O)_1$	$\sigma(W/O)_2$	$\sigma(W/O)_3$	$\sigma(W/O)_4$	3	6
A4	I3	I3	I4	I4	4F4.2				4F6.3				4F4.2					

CAMS Error Data Card 4

5	9	12	15	19	23	27	31	35	39	45	51	57	63	79	80
					Std. Dev. Proportion Est.				Bias in Proportion Estimate						
Country	Reg.	Zone	Strata	Sub-strata	$\sigma(P/W)_1$	$\sigma(P/W)_2$	$\sigma(P/W)_2$	$\sigma(P/W)_3$	$B(P/W)_1$	$B(P/W)_2$	$B(P/W)_3$	$B(P/W)_4$		4	6
A4	I3	I3	I4	I4	4F4.2				4F6.3						

SEE Data Cards (cont'd)

Signature Extension Data Card 1

5	9	12	15	21	27	31	35	39	43	47	51	55	59	63	67	71	75	79	80
Country	Reg.	Zone	B_{1W}	B_{2W}	σ_{11W}	σ_{12W}	σ_{13W}	σ_{14W}	σ_{15W}	σ_{16W}	σ_{21W}	σ_{22W}	σ_{23W}	σ_{24W}	σ_{25W}	σ_{26W}		1	7
A4	I3	I3	F6.3	F6.3	Mult. Std. Dev. Error W/W 6F4.2						Additive Std. Dev. Error W/W 6F4.2								

Signature Extension Data Card

5	9	12	15	21	27	31	35	39	43	47	51	55	59	63	67	71	75	79	80
Country	Reg.	Zone	B_{1M}	B_{2M}	σ_{11M}	σ_{12M}	σ_{13M}	σ_{14M}	σ_{15M}	σ_{16M}	σ_{21M}	σ_{22M}	σ_{23M}	σ_{24M}	σ_{25M}	σ_{26M}		2	7
A4	I3	I3	F6.3	F6.3	Mult. Std. Dev. Error W/M 6F4.2						Additive Std. Dev. Error W/M 6F4.2								

Signature Extension Data Card 3

5	9	12	15	21	27	31	35	39	43	47	51	55	59	63	67	71	75	79	80
Country	Reg.	Zone	B_{1O}	B_{2O}	σ_{11O}	σ_{12O}	σ_{13O}	σ_{14O}	σ_{15O}	σ_{16O}	σ_{21O}	σ_{22O}	σ_{23O}	σ_{24O}	σ_{25O}	σ_{26O}		3	7
A4	I3	I3	F6.3	F6.3	Mult. Std. Dev. Error W/O 6F4.2						Additive Std. Dev. Error W/O 6F4.2								

Signature Extension Data Card 4

5	9	12	15	21	27	31	35	39	43	47	51	55	59	63	67	71	75	79	80
Country	Reg.	Zone	B_1	B_2	σ_{11}	σ_{12}	σ_{13}	σ_{14}	σ_{15}	σ_{16}	σ_{21}	σ_{22}	σ_{23}	σ_{24}	σ_{25}	σ_{26}			
A4	I3	I3	F6.3	F6.3	Multiplicative Std. Dev. Error 6F4.2						Additive Std. Dev. Error 6F4.2								

SEE Data Cards (cont'd)

2.1.4 Rules for Entering Data Items on Cards

2.1.4.1 General

1. Integers must be right justified.
2. Alphanumeric data must be left justified.
3. F format numbers must have the decimal point present, i.e., F6.2 - NNN.NN. However, the user may override the specified field format as long as the total field width is not exceeded.
4. Except for IYESR, ISIGEX and ICAMER, all other nominal values are zero.

2.1.4.2 Rules for Inputting Specific Fields

- ICSESH must match the case no. of the substrata historical file.
- If IDFRS and IDTOS are left blank, then it will be assumed that the occurrence of any fatal error will terminate the writing of that output file with which it is associated.
- If the first entry of IERDTE data is 0 or blank on card 1 of a YES data group, then the card 2 data is ignored (the card 2 must be present but can be blank except for ID). The IERDTE data and card 2 data from the last non-zero IERDTE data group is used to obtain values for the current group.
- All standard deviation, bias and frequency data items are checked to be within valid ranges. See Section 5.2 for specific field checks.

2.2 FILE INPUT

The only input binary data file is the substrata historical file. See Section 2.4 of the Users Manual for format and contents.

3.0 PROCESSING

3.1 OVERVIEW

The program is basically an input data card error checker and data file generator.

The program will be organized into five basic routines consisting of driver, YES input data processor, CAMS input data processor, signature extension input data processor and error processor. The driver controls the reading of the substrata historical file and the calling of the input processors. Each of the input processors reads an input card group, checks each data field for errors, prints on option the input card image, correlates the substrata historical file with the input data and writes an output file. The error processor catalogs and prints out the input data error messages. The program will continue to read each data type and error check the data until an EOF is reached for that data, regardless of how many errors are detected. If a fatal occurs for a data type, the writing of its output file is terminated. The only exception to this process is that if any input data set is out of order, the program terminates processing and returns control to the system.

3.2 PROCEDURES AND EQUATIONS

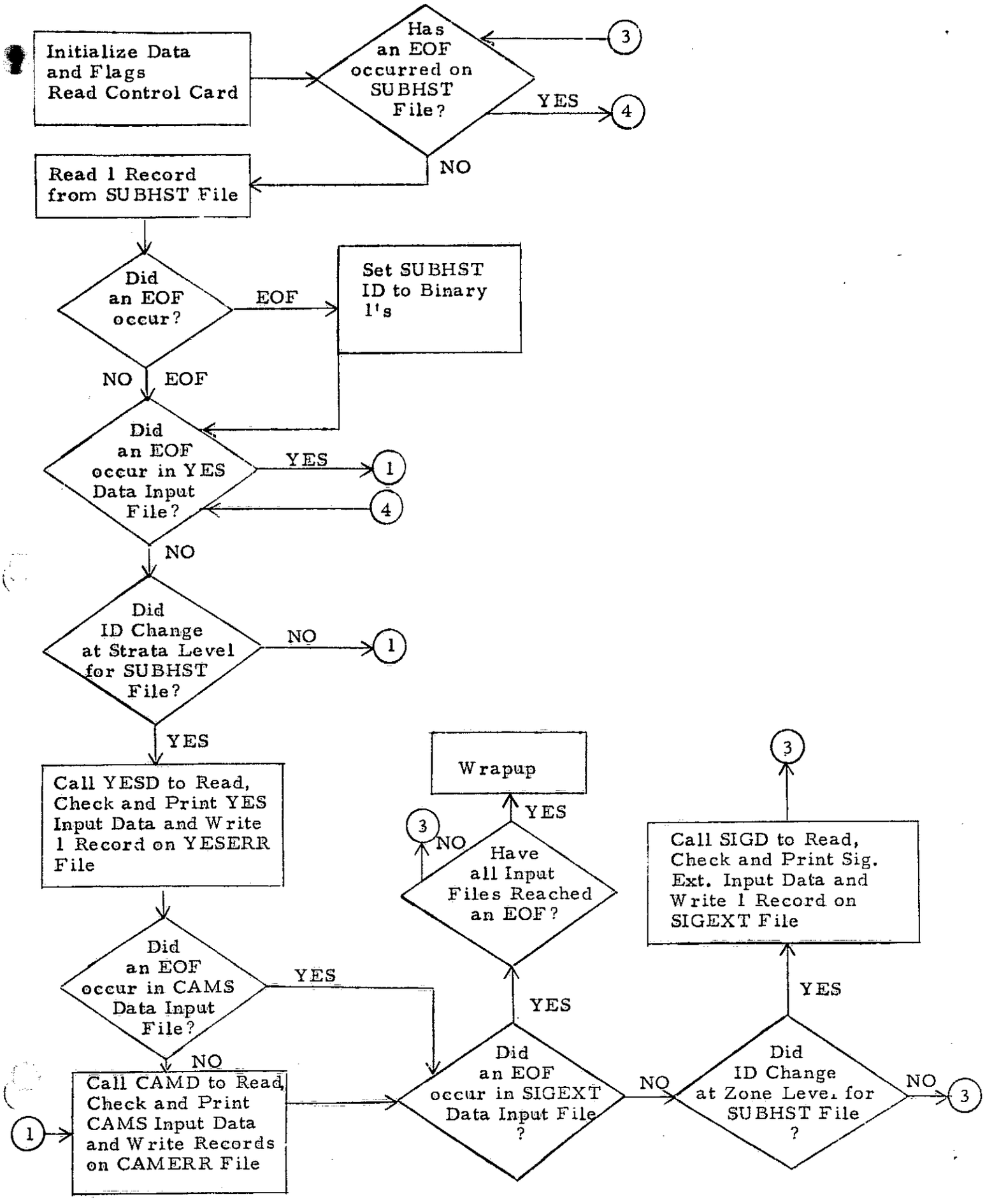
There are no application oriented procedures performed except for error checking which is described in Section 5.0.

The only calculation performed is that if the country is either USA or CAND, then the yield data is input in bushels per acre and must be converted to quintals per hectar before being written to the YESERR file. The conversion equations are:

1. YIELD = YIELDI / 1.4869664
2. BIASY(I) = BIASYI(I) / 1.4869664 for $1 \leq I \leq 6$
3. SIGYIE(I) = SIGYII(I) / 1.4869664 for $1 \leq I \leq 6$

In addition, the truncation date IERDTE is converted to Zulu date prior to being written on the YESERR file.

Block Diagram



4.0 OUTPUT

There are four media of output:

1. PRINT file
2. YESERR file
3. CAMERR file
4. SIGEXT file

4.1 PRINT DATA

4.1.1 ECHO Print

If ILIST = 1 or 2, then each input data card image will be listed in a form as close to the input form as possible.

Regardless of ILIST's value, if an error is found on a card then the card image is printed in an identical manner.

Since each input data file is processed alternately, the listing of each card type will be intermixed with each other.

Print Format for the Control Card

		IDFRS		IDTOS					
ICASE	ILIST	REG	ZONE	REG	ZONE	IYESR	ISIGER	ICAMR	ICSESH
NNNN	N	NNN	NNN	NNN	NNN	NN	NN	NN	NNNN

Print Format for Bulk Card Data

YES Data Group

←	CARD IMAGES FOR 1 ID GROUP	→ 15
←	CARD IMAGES FOR 1 ID GROUP	→ 25

CAMS Data Group

←	CARD IMAGES	→ 16
	↓	26
		36
		46

Sig Ext Data Group

←	CARD IMAGES	→ 17
	↓	27
		37
		47

Repeat for each type and group unless a particular type is omitted via the file unit flag.

4.1.2 Intermediate Debug Print

If ILIST = 2, the contents of each record written on an output file is printed. Again the printout for each file is intermixed in the order of processing. Assuming all three files are being written, the format is as follows:

YESERR File

ICTRY = AAAA IREG = NNN IZONE = NNN ISTRAT = NNNN
Y = NN.NN (IERDTE(I), BIASY(I), SIGYIE(I), I = 1, 6) =
NNNNN NNN.N NN.NN NNNNN ——————→ etc.

CAMERR File

ICTRY = AAAA IREG = NNN IZONE = NNN ISTRAT = NNNN ISUB = NNNN ISEG = NNNN
(PWW(I), PWM(I), PWO(I), BWW(I), SIGWW(I), BWO(I), SIGWO(I),
BWM(I), SIGWM(I), BPW(I), SIGPW(I), I = 1, 4) =
N.NN N.NN N.NN NN.NNN N.NN NN.NNN N.NN NN.NNN N.NN NN.NNN N.NN

Second set is on same line.
Third and 4th set is on 5th line.

SIGEXT File

ICTRY=AAAA IREG = NNN IZONE = NNN
B1W = NN.NNN B2W = NN.NNN SIG1W = $N.NN_1$ --- $N.NN_6$ SIG2W = $N.NN_1$, -- $N.NN_6$
B1M = NN.NNN B2M = NN.NNN SIG1M = $N.NN_1$ --- $N.NN_6$ SIC2M = $N.NN_1$, -- $N.NN_6$
B1O = NN.NNN B2O = NN.NNN SIG1O = $N.NN_1$ --- $N.NN_6$ SIG2O = $N.NN_1$, -- $N.NN_6$
B1 = NN.NNN B2 = NN.NNN SIG1 = $N.NN_1$ --- $N.NN_6$ SIG2 = $N.NN_1$ --- $N.NN_6$

4.1.3 Status Information

The following information will print out at the end of the run.

NNNN SUBSTRATA HISTORICAL FILE RECORDS READ
NNNN YES DATA CARDS READ
NNNN CAMS ERROR DATA CARDS READ
NNNN SIGNATURE EXTENSION DATA CARDS READ
NNNN RECORDS WRITTEN ON YESERR FILE. ICSEYM = NNNN
NNNN RECORDS WRITTEN ON CAMERR FILE. ICSECE = NNNN
NNNN RECORDS WRITTEN ON SIGEXT FILE. ICSESE = NNNN
NNN INPUT ERRORS DETECTED

Either one or the other message will print

JOB COMPLETED

FATAL ERROR OCCURRED IN SEE, FOLLOWING FILES MAY NOT BE ANY GOOD

AAAAAA AAAAAA AAAAAA

4.2 FILE OUTPUT

Up to three files are output from this program.

1. YESERR - YES Error File
2. CAMERR - CAMS Error File
3. SIGEXT - Signature Extension File

See Section 2.4 of the Users Manual for contents and format.

5.0 ERROR PROCESSING

With the exception of the wrong substrata historical file being mounted or any data out of sequence, the program will continue to check for all types of errors even though the writing of a given output file has stopped. There are three categories of errors:

1. Fatal with immediate termination - level 3.
2. Fatal for a given input data type - error checking continues but the writing of the corresponding output file stops - level 2.
3. Nonfatal - normal processing continues - level 1. The card image of the card with field errors or card ID which triggered a general error will be printed prior to the message. The error checks, messages and fatal status are given in Sections 5.1 and 5.2 below.

5.1 GENERAL

Level 3
Code No.

1. A check is made that the correct substrata historical file has been mounted as specified by ICSESH.

   Message:

   EITHER THE SUBSTRATA HISTORICAL FILE HAS NOT BEEN MOUNTED OR IT HAS THE WRONG CASE NO. PROCESSING IS TERMINATED.

2. A check on sequence is made from country down to the card sequence no. for each active card input set. In addition, a missing sequence no. in a card group with the same ID will cause the same action to be taken.

   Message:

   THE ------ INPUT DATA SET IS OUT OF ORDER. PROCESSING IS TERMINATED.

3. A check is made to make sure the ID in IDTOS is ≥ the ID in IDFRS.

   Message:

   ID IN IDTOS IS NOT GREATER OR = TO IDFRS

The following errors are level 2 if the error occurs when the current ID is in the range specified by IDFRS and IDTOS. Otherwise, it is a level 1 error.

10. A check is made to make sure that each SUBHST record has a corresponding record for each of the active input card sets for YES and Sig. Ext. This situation will only be considered an error in case of Sig. Ext. input if there is at least one segment in the SUBHST record.

    Message:

    Optional line
    LEVEL 2 - WRITING OF ------ FILE TERMINATED
    ICTRY ---- IREG ---- IZONE ---- ISTRAT ---- ISUB ----
    ------ INPUT DATA SET DOES NOT HAVE A RECORD WHICH CORRESPONDS TO SUBHST ID.

11. A check is made to make sure that for each record in SUBHST with one or more sample segments there is a CAMS input card group for each segment.
    Message:
    ICTRY ---- IREG --- IZONE --- ISTRAT ---- ISUB ----
    CAMS INPUT DATA SET DOES NOT EXIST FOR SEGMENT ----
    IN SUBHST RECORD.

Level 1 Error
Code No.

15. A check is made to see if an extra input card group has been input with no match on the SUBHST file.

    Message:

    THE ------ INPUT GROUP DOES NOT MATCH WITH SUBHST FILE AND IS EXTRA DATA

5.2 CARD FIELD ERRORS

The occurrence of any of these errors is level 2 if the error occurs when the current ID is in the range specified by IDFRS and IDTOS. Otherwise, it is considered to be a level 1 error.

Code No.

20. A check is made to make sure the error truncation dates in IERDTE on YES card 1 are in ascending order (if not all 0).

    Message:

    IERDTE ARRAY HAS DATES NOT IN ASCENDING ORDER

21. A check is made to make sure YIELDI on YES card 1 is between 0 and 99.99.

    Message:

    YIELDI IS NOT BETWEEN 0 AND 99.99

22. A check is made to make sure that BIASYI on YES card 2 is between -99.9 and +99.9.

    Message:

    ABSOLUTE VALUE OF BIASYI IS NOT LESS THAN 99.9

23. A check is made to make sure that SIGYII on YES card 2 is between 0 and +99.9.

    Message:

    SIGYII IS NOT BETWEEN 0 AND 99.9

24. A check is made to make sure that the P(W/i) data for a CAMS card group is between 0 and 1.

    Message:

    PW- IS NOT BETWEEN 0 AND 1

25. A check is made to make certain that B(W/i) and B_{PW} for a CAMS card group is between -9.999 and +9.999.

    Message:

    ABSOLUTE VALUE OF BW- OR BPW IS NOT LESS THAN 9.999

Code No.

26. A check is made to make certain that σ_{PW} for a CAMS card group is between 0 and 9.99.

    Message:

    SIGW- OR SIGPW IS NOT BETWEEN 0 AND 9.99

27. A check is made to make sure that B_{1i} and B_{2i} data on the signature extension card group is between -9.999 and +9.999.

    Message:

    ABSOLUTE VALUE OF B1- OR B2- IS NOT LESS THAN 9.999

28. A check is made to be sure that σ_{1i} or σ_{2i} data on the signature extension card group is between 0 and 9.99.

    Message:

    SIG1- OR SIG2- IS NOT BETWEEN 0 AND 9.99

29. A check is made to make sure the dates in IERDTE are in range. Year must be greater than 64, month must be between 1 and 12 and day must be between 1 and 31.

    Message:

    IERDTE ARRAY HAS A BAD YEAR, MONTH OR DAY NUMBER

PART II

COMMON BLOCK DEFINITIONS

COMMON STORAGE ALLOCATION

Name FLAG Size Page 1 of 4

Function

Name	Dimen-sion	For-mat	Description	Sym-bol	Units
ICTRSV					
IREGSV			Saved SUBHST ID		
IZONSV					
IOUT			Unit 6		
INP			Unit 5		
ICASE	1	I	See Problem Description Case No.		
ILIST	1	I	0 - List cards in error only 1 - Echo input card image 2 - Echo input and print output records		
IDFRS	2	I	Starting region + zone for level 2 errors to occur		
IDTOS	2	I	Ending region + zone for level 2 errors to occur		
IYESU	1	I	Input YES unit, nominal value = 1 (- value indicates do not read)		
ISIGEX	1	I	Input Sig. Ext. unit, nominal value = 2 (- value indicates do not read)		
ICAMER	1	I	Input CAMS unit, nominal value = 4 (- value indicates do not read)		
ICSESH	1	I	Case no. of substrata historical file		
ICTRY	1	I	SUBHST ID for current record Country		
IREG	1	I	Region		
IZONE	1	I	Zone		
ISTRAT	1	I	Strata		

COMMON STORAGE ALLOCATION

Name FLAG Size Page 2 of 4

Function

Name	Dimen-sion	For-mat	Description	Sym-bol	Units
ISUB	1	I	SUBHST ID for current record Substrata		
IREGP	1	I	Previous SUBHST record ID Region		
IZONEP	1	I	Zone		
ISTRAP	1	I	Strata		
IFATER	1	I	≠0 indicates a level 3 error has occurred and job is abandoned		
IZSGC	1	I	Count of no. of segments in a zone		
ITYPE	1	I	1 - YES error, 2 - CAMS error, 3 - Sig. Ext. error, error type		
IFATEY	1	I	≠0 indicates a level 2 error has occurred for a particular input - YES input data		
IFATEC	1	I	CAMS error input data		
IFATES	1	I	Signature extension input data		
IRANGE	1	I	=0, no level 2 error can occur ≠0, SUBHST ID in range for level 2		
ICAMR	1	I	=0, read CAM input file		
IYESR	1	I	≠0, don't read YES input file		
ISIGR	1	I	Sig. Ext. input file		
IERR	1	I	Count on no. of errors which have occurred		
IPRINT	1	I	=0, print card image for this error ≠0, card image already printed		
IEFSB	1	I	=0, end of file has not occurred on SUBHST ≠0, EOF has occurred		
ISBHS	1	I	Substrata history file unit no. ≡ 3		

COMMON STORAGE ALLOCATION

Name FLAG Size Page 3 of 4

Function

Name	Dimen-sion	For-mat	Description	Sym-bol	Units
IERCDE	1	I	Error code no. as defined in Section 5.2, Problem Description		
ISUBCT	1	I	No. of records read from SUBHST file		
IYSCT	1	I	No. of YES data group records read in		
ICMCT	1	I	No. of CAMS data group records read in		
ISGCT	1	I	No. of Sig. Ext. group records read in		
IYSERC	1	I	No. of records written on YESERR file		
ICMERC	1	I	No. of records written on CAMS error file		
ISIGEC	1	I	No. of records written on Sig. Ext. file		
ILEVES	1	I	Save location for substrata file change level indicator - ILEVEL. Used by SIGD.		
IPSEG	1	I	Seg. ID on SUBHST file with no CAMS input rec. for use by ERRPRO		
IYESER	1	I	YES error output unit ≡ 11		
ICAME	1	I	CAMS error output unit ≡ 8		
ISIGE	1	I	Sig extension output unit ≡ 9		
IREG2P	1	I			
IZON2P	1	I			
ISTR2P	1	I	Previous ID for CAMS input file		
ISUB2P	1	I			
ISEG2P	1	I			

COMMON STORAGE ALLOCATION

Name FLAG Size Page 4 of 4

Function

Name	Dimen-sion	For-mat	Description	Sym-bol	Units
IREG1P	1	I			
IZON1P	1	I	Previous ID for YES input file		
ISTRIP	1	I			
IREG3P	1	I			
IZON3P	1	I	Previous ID for Sig. Ext. input file		
ILEVEL	1	I	=1 - highest ID level chg substr. =2 - highest ID level chg strata =3 - highest ID level chg zone =4 - highest ID level chg region for SUBHST file		
IFILT1	1	I	For CAMS error data input =0 - constant for segment or none =1 - constant for substrata on down =2 - constant for strata on down =3 - constant for zone on down =4 - constant for region on down		
IFILT2	1	I	For Sig. Ext. same meaning except 0 - means none; 3 - zone; 4 - region		

COMMON STORAGE ALLOCATION

Name SIGIN Size _____ Page 1 of 2

Function Contains data for one signature extension input/output group

Name	Dimension	Format	Description	Symbol	Units
ICTRY3	1	I	Country		
IREG3	1	I	Region		
IZONE3	1	I	Zone		
B1W	1		See Problem Description		
B2W	1				
SIG1W	6				
SIG2W	6				
B1M	1				
B2M	1				
SIG1M	6				
SIG2M	6				
B1O	1				
B2O	1				
SIG1O	6				
SIG2O	6				
B1	1				
B2	1				
SIG1	6				

COMMON STORAGE ALLOCATION

Name SIGIN Size _____ Page 2 of 2

Function _____

Name	Dimen-sion	For-mat	Description	Sym-bol	Units
SIG2	6		See Problem Description		

COMMON STORAGE ALLOCATION

Name CAMSIN Size _____ Page 1 of 1

Function Contains data for one CAMS input/output group

Name	Dimen-sion	For-mat	Description	Sym-bol	Units
ICTRY2	1	I	See Problem Description		
IREG2	1	I			
IZONE2	1	I			
ISTRA2	1	I			
ISUB2	1	I			
ISEG2	1	I	Segment ID		
PWN	4	Flt.			
PWM	4				
PWO	4				
BWW	4				
BWM	4				
BWO	4				
SIGWW	4				
SIGWM	4				
SIGWO	4				
BPW	4				
SIGPW	4				

COMMON STORAGE ALLOCATION

Name YESIN Size _____ Page 1 of 1

Function Contains data for one YES input/output group with = ID

Name	Dimen-sion	For-mat	Description	Sym-bol	Units
ICTRY1			Country		
IREG1			Region		
IZONE1			Zone		
ISTRA1			Strata		
YIELDI	1	Flt.	YIELD - See Problem Description		
IERDTE	3x6	I	Year, month and day for six error truncation dates; see Problem Description		
BIASYI	6	Flt.	YIELD bias error for dates; see Problem Description		
SIGYII	6	Flt.	YIELD std. dev. for dates; see Problem Description		
YIELD	1	Flt.	YIELD for YESERR file	Y	Quintals/Hectare
ITRDTE	6	I	Truncation date in Zulu for YESERR file	-	Days
BIASY	6	Flt.	YIELD bias for YESERR file	B_Y	Quintals/Hectare
SIGYIE	6	Flt.	YIELD std. dev. for YESERR file	σ_Y	Quintals/Hectare

COMMON STORAGE ALLOCATION

Name SUBHIS Size _____ Page 1 of 1

Function Detail record from substrata historical file; also header record data

Name	Dimen-sion	For-mat	Description	Sym-bol	Units
NSEG	1	I	No. of sample segments in this substrata		
IDSEG	150	I	List of segment no's. 0 terminates.		
IGRP	1	I			
PWHTH	1	Flt.			
AREA	1	Flt.			
PWTHT	1	Flt.			
NAGR	1	I			
NAL	1	I			
DELPW	1	Flt.			
DELPM	1	Flt.			
CV	(4)	Flt.			
NAME	2	I	8 character name of SUBHST file		
ICSE	1	I	Case no. of SUBHST file		
IMXSEG	1	I	Maximum no. of segment no's in IDSEG		

PART III

LIST OF SUBROUTINES AND SUBROUTINE CALL STRUCTURE

C-4

LIST OF SUBROUTINES IN SEE

Name	Function
SEE	Main control routine for the program. Performs initialization and wrapup functions.
SEECON	Reads the SUBHST file and controls the reading and processing of the three input data files.
YESD	Reads a YES data card group, checks the data for errors and writes a record on the YESERR file.
SIGD	Reads a signature extension ID group and checks for errors and writes the record on the CAMERR file.
CAMD	Controls the reading of CAMS input data groups for all segments with = ID to SUBHST file ID.
SEGPRO	Reads one CAMS error input ID group into CAMSIN COMMON and checks it for errors.
ERRPRO	Controls the processing of all errors, prints error messages and card images according to error type.
LFPA	Given month, day and year, computes Zulu date.

SUBROUTINE CALL STRUCTURE

```
SEE
    ERRPRO
    SEECON
        YESD
            LFPA
            ERRPRO
        CAMD
            ERRPRO
            SEGPRO
                ERRPRO
        SIGD
            ERRPRO
```

PART IV

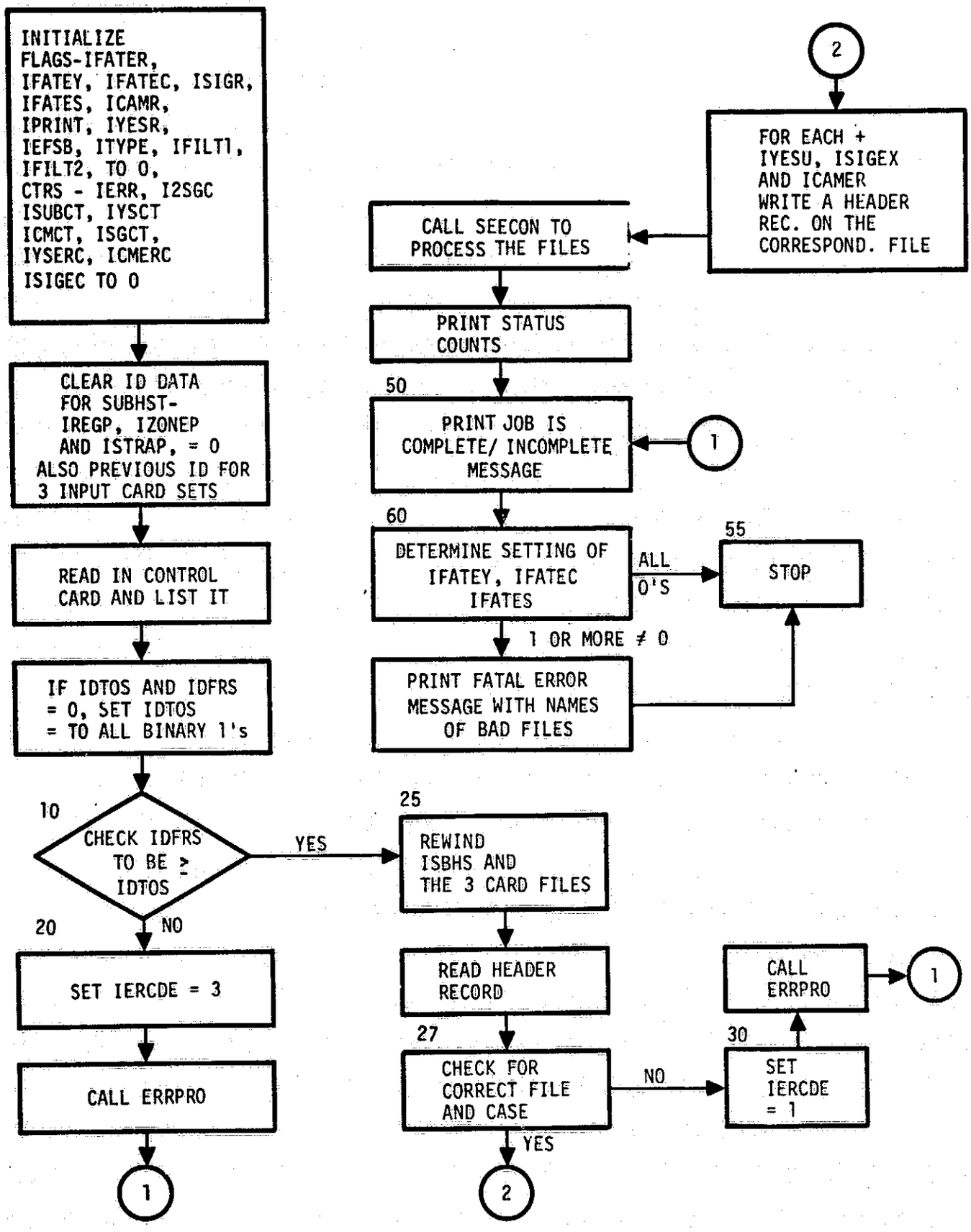
SUBROUTINE DESCRIPTIONS AND FLOWCHARTS

SEE MAIN ROUTINE

Purpose:

Provide main control for the SEE program. It initializes data, reads in control card, lists and checks it, checks for correct substrata historical file to be mounted, calls the input file processing routine SEECON and prints job status information.

MAIN ROUTINE FLOWCHART



SUBROUTINE SEECON

Purpose:

This subroutine reads the substrata historical file and controls the reading of the three input data files via subroutine calls. It determines when and if a particular input processor subroutine is to be called.

Input:

FLAG COMMON
IYESU, ISIGEX, ICAMER, ICTRY, IREG, IZONE, ISTRAT, ISUB, IREGP, IZONEP, ISTRAP, IFATEY, IFATEC, IFATES, IEFSB, ISBHS, ISUBCT, IDFRS, IDTOS

SUBHIS COMMON
NSEG, IDSEG, IMXSEG

Output:

FLAG COMMON
ICTRY, IREG, IZONE, ISTRAT, ISUB, IZSGC, IRANGE, LZONE, LSTRAT, IPRINT, IEFSB, IERCDE, ISUBCT

Linkage:

CALL SEEPRO

Subroutines Used:

CALL YESD
CALL CAMD
CALL SEGD
CALL ERRPRO

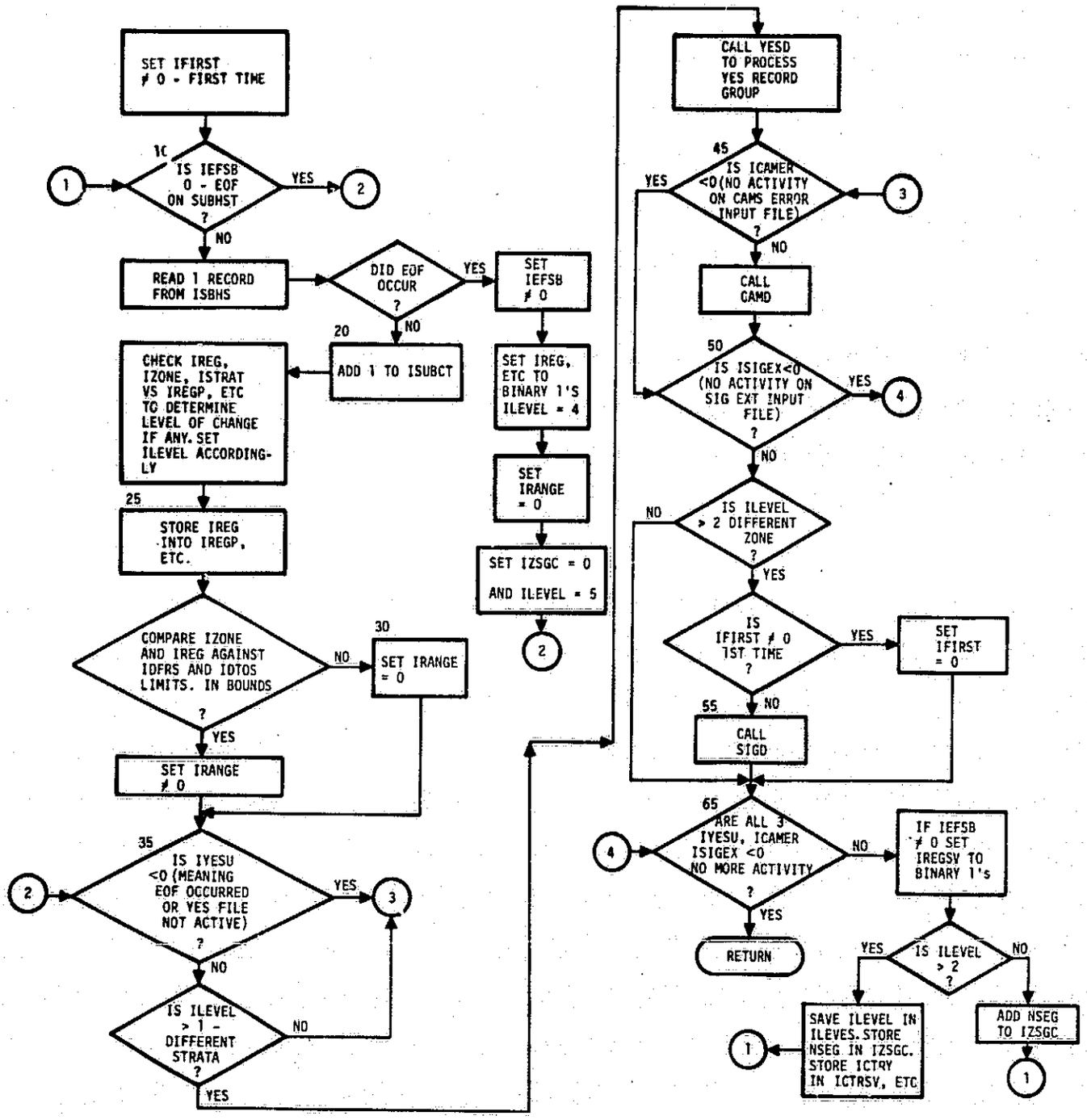
Local Variable Description:

IFIRST - First time flag
≠0 - first time, =0 - not first time

Processing:

See flowchart.

SUBROUTINE SEECON FLOWCHART



SUBROUTINE YESD

Purpose:

To read a YES record group, list the card images on option, check the data for field errors, check for correct order, determine if current YES ID is compatible to ID of SUBHST, write a record on the YESERR file and indicate when processing for this file is complete. This routine will allow duplicate data groups for any ID level to be entered only once.

Input:

FLAG COMMON
ICTRY, IREG, IZONE, ISTRAT, IFATEY, IYSCT, IYSERC, ILIST, IYESR, IYESER, IYESU, IREG1P, IZON1P, ISTR1P

YESIN COMMON
ICTRY1, ISTRA1, YIELDI, IERDTE, BIASYI, SIGYII

Output:

YESIN COMMON
ID and YIELD, ITRDTE, BIASY, SIGYIE

FLAG COMMON
ITYPE, IPRINT, IERCDE, IYSCT, IYSERC, IYESR, IREG1P, IZON1P, ISTR1P

Linkage:

CALL YESD

Subrouitnes Used:

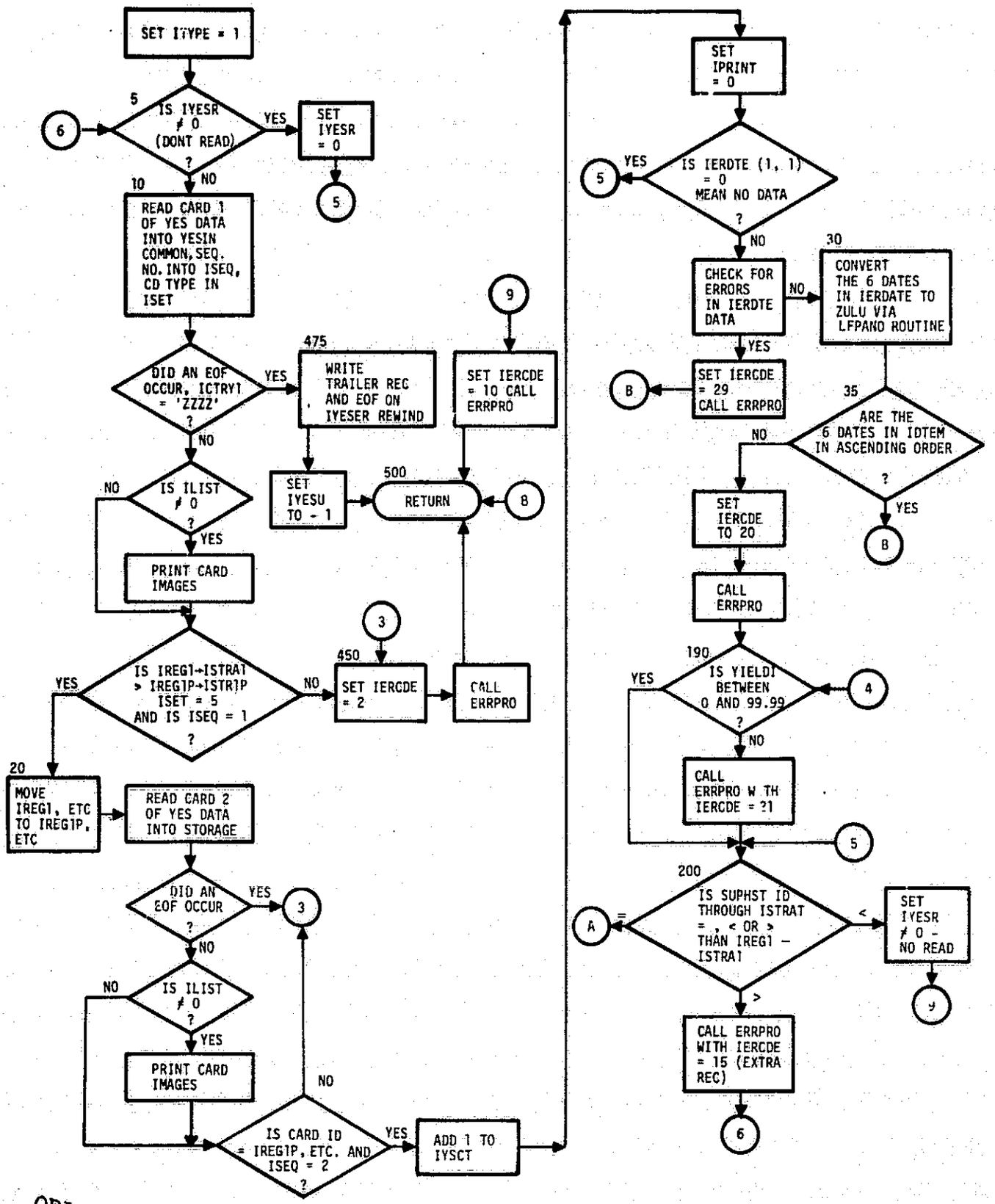
CALL LFPA (day, month, year, dum, days)
Given day, month, year, compute days in Zulu.
CALL ERRPRO

Local Variable Descriptions:

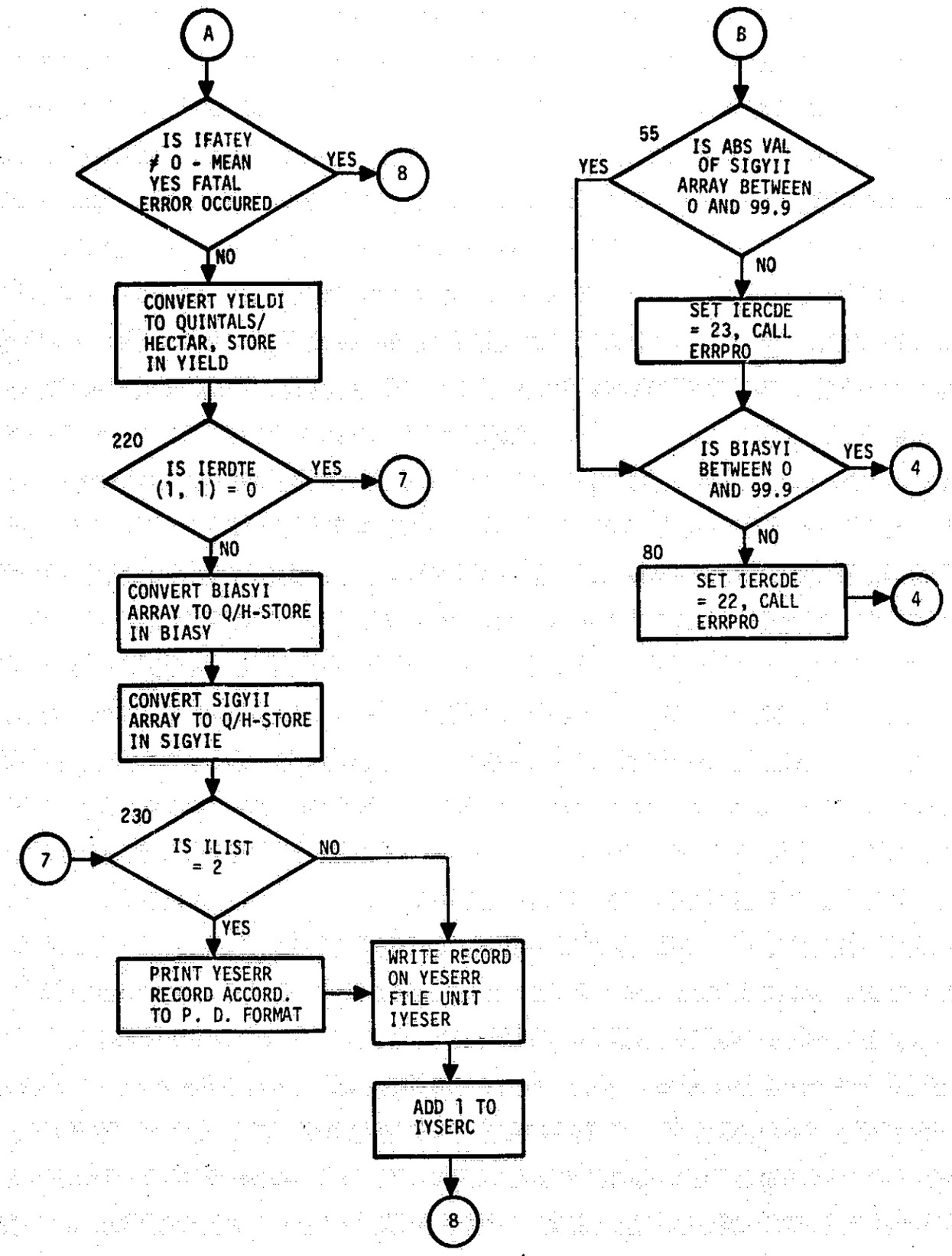
ISEQP	Previous sequence no. for cards
ISEQ	Current card seq. no.
ISET	Card set no. should ≡ 5
IDTEM(6)	Temporary Zulu IERDTE loc.

Processing: See flowchart.

YESD FLOWCHART



YESD FLOWCHART (CONTINUED)



SUBROUTINE SIGD

Purpose:

This routine reads a signature extension record group, lists the card images on option, checks the data for field errors and correct order. It determines if SIG. EXT. ID is compatible to SUBHST ID, writes a record on the SIGEXT file and indicates when processing for this file is complete. This routine will allow duplicate data groups for any ID level to be entered once.

Input:

FLAG COMMON
ICTRSV, IREGSV, IZONSV, IFATES, ISGCT, ISIGEC, ILIST, ISIGR, ISIGEX, ISIGE, ILEVEL, IFILT2, IZSGC, IREG3P, IZON3P

SIGIN COMMON
ICTRY3, IREG3, IZONE3, plus all signature extension data.

Output:

FLAG COMMON
ITYPE, IPRINT, ISGCT, ISIGEC, IREG3P, IZON3P, IERCDE, ISIGR

SIGIN COMMON
All data.

Linkage:

CALL CAMD

Subroutines Used:

CALL ERRPRO

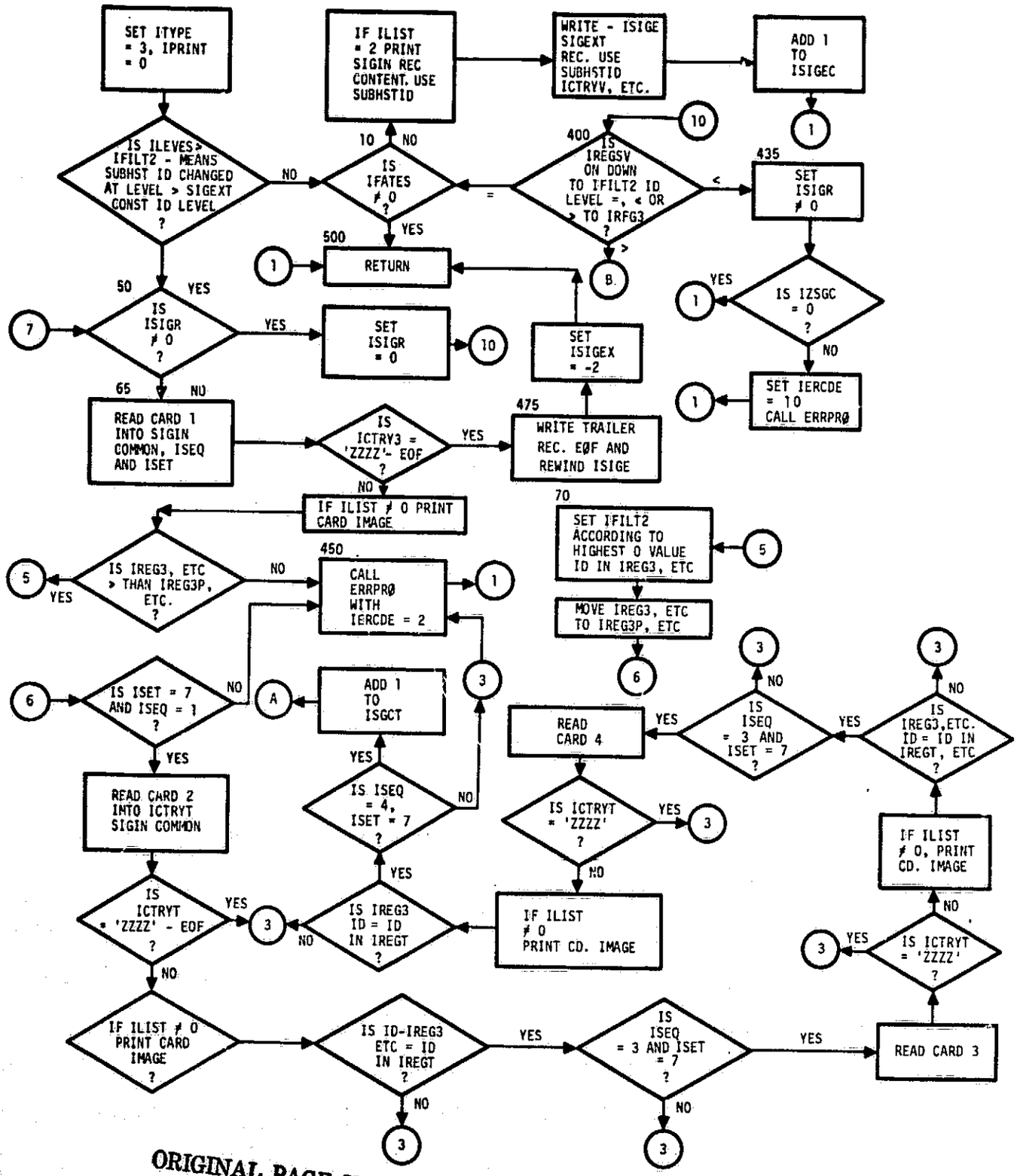
Processing:

See flowchart.

Local Variable Descriptions:

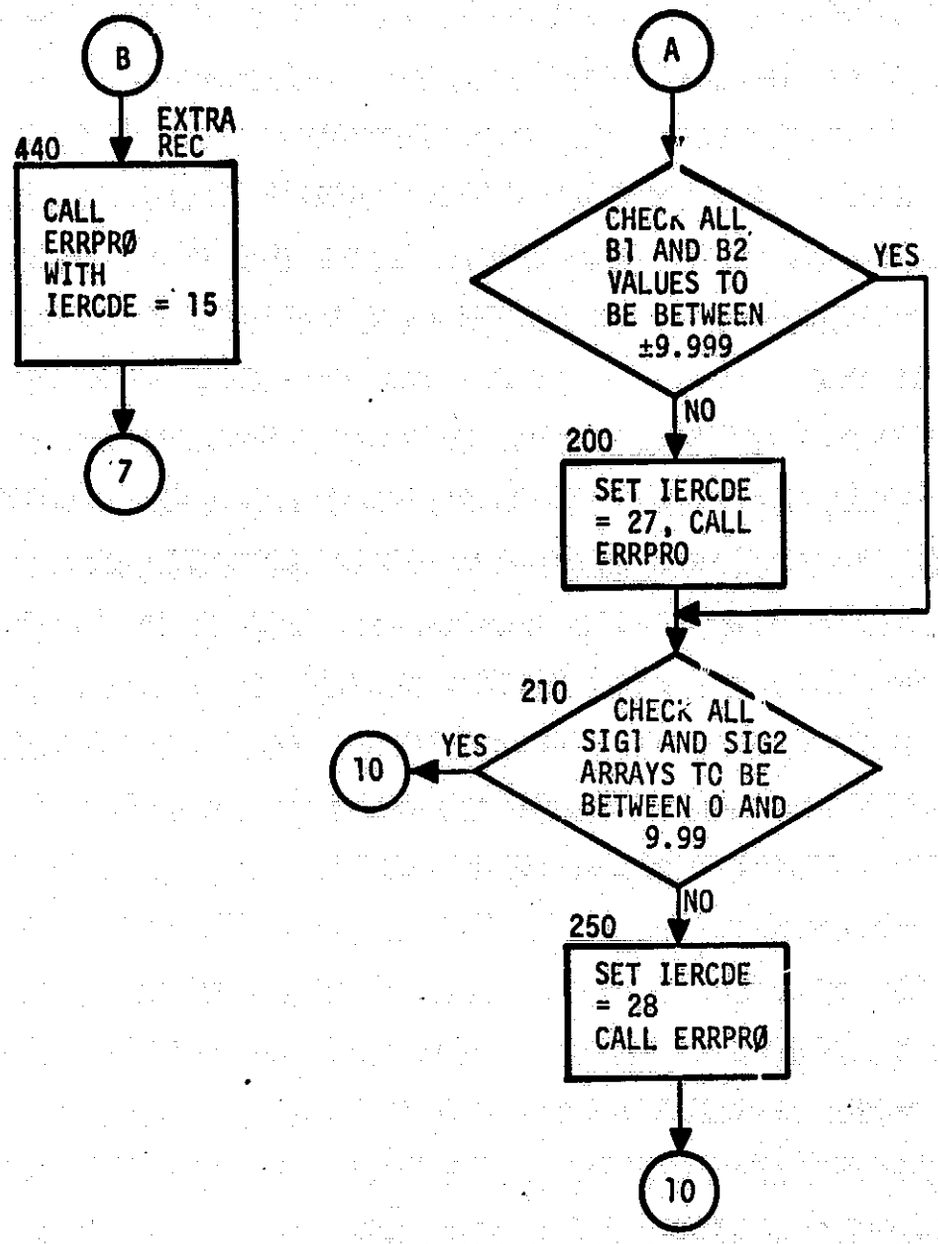
ICTRYT, IREGT, IZONET - temporary ID locations.

SIGD FLOWCHART



ORIGINAL PAGE IS OF POOR QUALITY

SUBROUTINE SIGD FLOWCHART (CONT'D)



SUBROUTINE CAMD

Purpose:

To control the reading of CAMS input data groups for all segments with = ID to SUBHST file. Check for data to be in order and not missing; also check for ID compatibility with SUBHST file (accounting for multilevel constant data). Make sure there is a segment for each segment in SUBHST file substrata record. Write a record on the CAMERR file and indicate when processing for this file is complete.

Input:

FLAG COMMON
ICTRY, IREG, IZONE, ISTRAT, IFATEC, ICAMR, ICMERC, ILEVEL, IFILT1, ICAME, ILIST, ICAMER, IREG2P, IZON2P, ISTR2P, ISUB2P, ISEG2P

CAMSIN COMMON
ICTRY2, IREG2, IZONE2, ISTRA2, ISUB2, ISEG2, and all application data

SUBHIS COMMON
NSEG, IDSEG

Output:

FLAG COMMON
IPSEG, ITYPE, IERCDE, ICMERC, ICAMR

CAMSIN COMMON
All data

Linkage:

CALL CAMD

Subroutines Used:

CALL SEGPRO (IND, IOPT) to read in 1 data group for a segment and do field/seq check.

IND =0 = SUBSTRA ID
≠0 ≠ SUBSTRA ID or EOF

IOPT =0 - Full process with ID change
≠0 - Ignore data with ID change

CALL ERRPRO

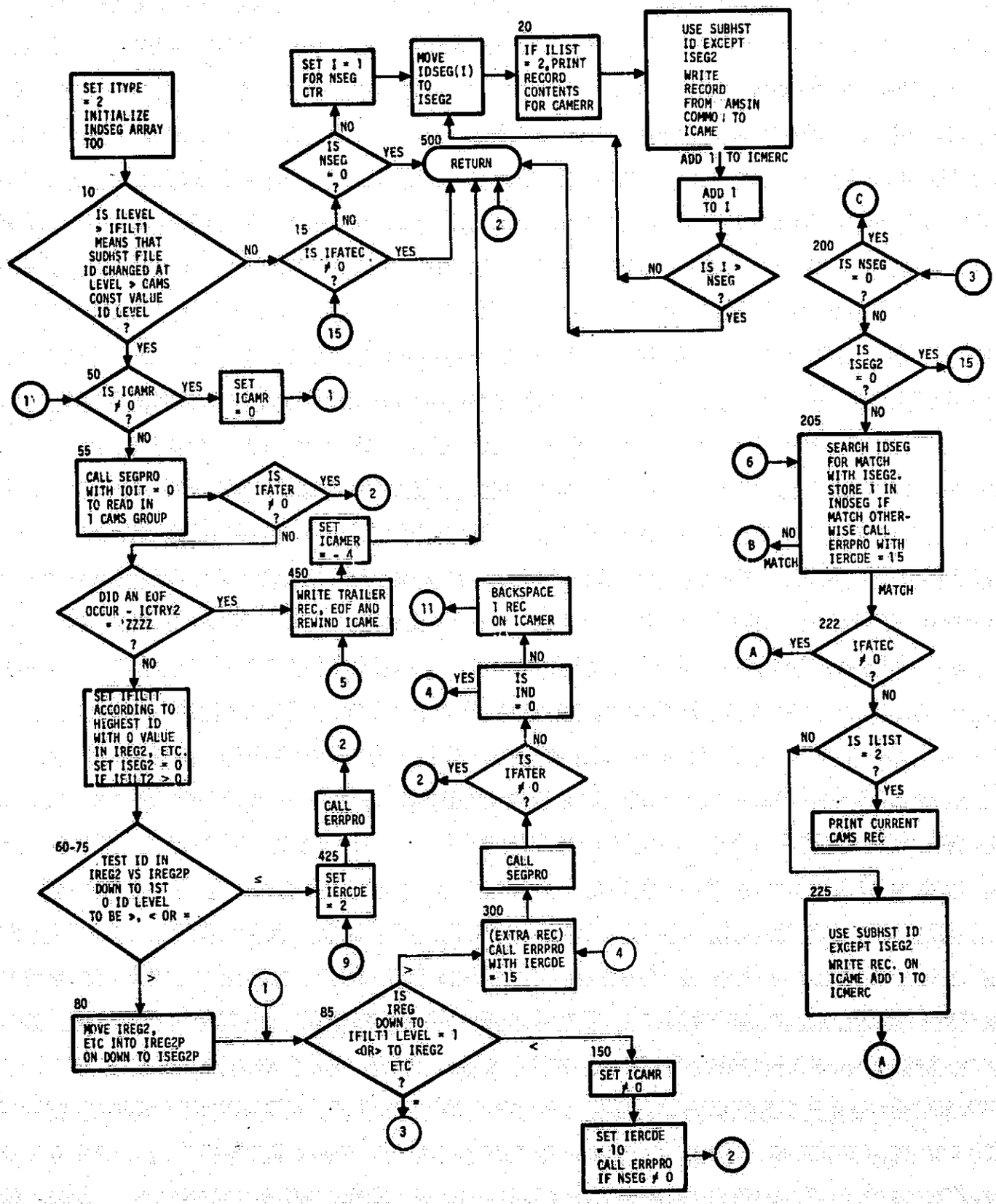
Processing:

See flowchart.

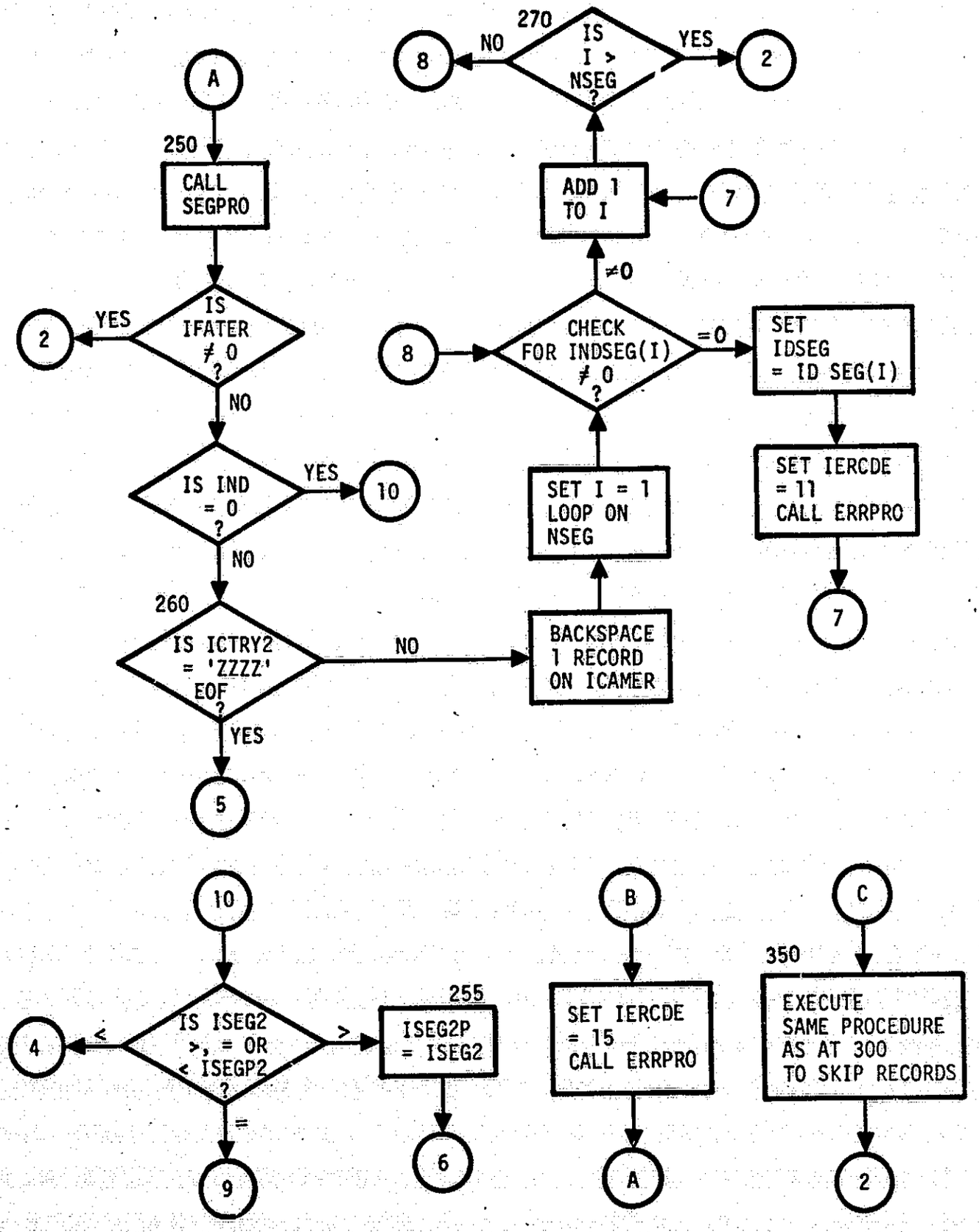
Local Variable Descriptions:

INDSEG (150) - Indicator array 1 to 1 with segment list in SUBHST record - 0 → no match on CAM input; ≠0 - means seg ID matched.

SUBROUTINE CAMD FLOWCHART



SUBROUTINE CAMD FLOWCHART (CONTINUED)



SUBROUTINE SEGPRO (IND, IOP)

Purpose:

To read 1 CAMS error input ID group into CAMSIN COMMON, determine if new ID has been read in; if so, set IND ≠ 0. If not, check order and completeness within group and perform field checks. Print card images of group if ILIST ≠ 0.

Input:

FLAG COMMON
ILIST, IREG2P, IZON2P, ISTRA2P, ISUB2P, ICAMER, ICMCT, ISEQ, ISET

CAMSIN COMMON

All data

Output:

CAMSIN COMMON
All data

FLAG COMMON
IPRINT, IERCDE, ICMCT

Linkage:

CALL SEGPRO (IND, IOPT)

On exit - IND = 0 No change in ID sort field
IND ≠ 0 Change in ID sort fields

On entry - IOPT = 0 Means an initial substrata call; full process if change in ID
IOPT ≠ 0 If ID change, set IND ≠ 0 and exit immediately

Subroutines Called:

CALL ERRPRO

Processing:

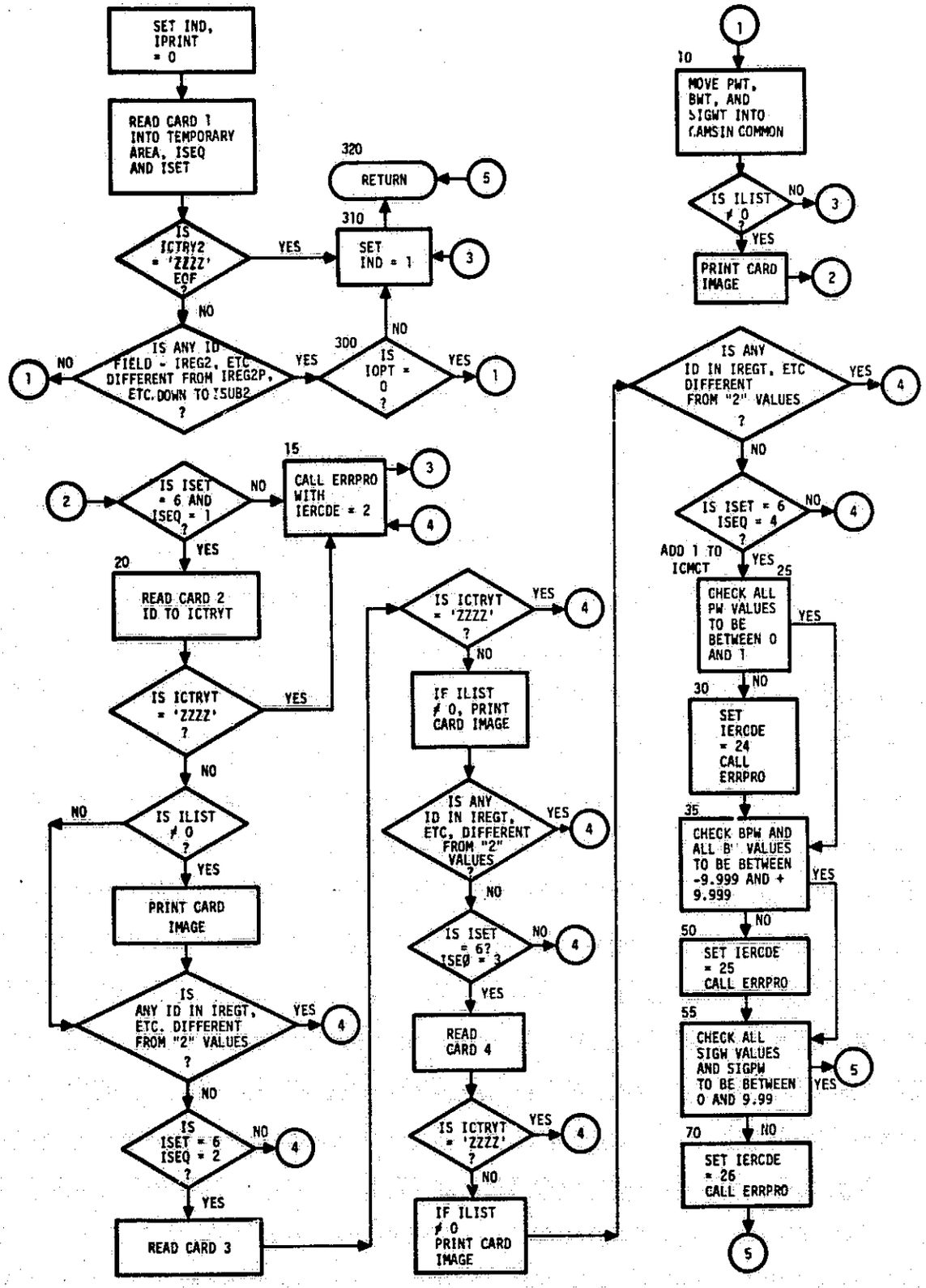
See flowchart.

Local Variable Descriptions:

IREGT, IZONET, ISTRTT, ISUBT, ISEGT, ICTRYT - temporary ID locations

PWT(4), BWT(4) and SIGWT(4) - temporary location for card 1 data

SUBROUTINE SEGPRO FLOWCHART



ORIGINAL PAGE IS
OF POOR QUALITY

SUBROUTINE ERRPRO

Purpose:

To process errors by category; setting status, initialing wrapup, printing messages and printing card images according to the category.

Input:

FLAG COMMON
IPSEG, IERR, IERCDE, IPRINT, ITYPE, IRANGE, ILIST, IYESER, ICAME, ISIGE, ICTRSV, IREGSV, IZONSV, ICTRY, IREG, IZONE, ISTRAT, ISUB

All data in CAMSIN and SIGIN COMMONS and all data except YIELD, ITRDTE, BIASY and ISGYIE in YESIN COMMON.

Output:

FLAG COMMON
IFATER, IFATEY, IFATEC, IFATES, IERR, IPRINT, IYESU, ISIGEX, ICAMER

Messages as described in problem description:
DATA PRINT OUT FOR ERROR CODES 2, 10-19 ASSUMING ILIST = 0, OTHERWISE JUST MESSAGE PRINTS.

Prints:
ICTRY ---- IREG --- IZONE --- ISTRAT ---- ISUB ---- ISEG ----

Message:
DATA PRINT OUT FOR ERROR CODES 20-28 ASSUMING ILIST = 0, OTHERWISE JUST MESSAGE PRINTS.

ICTRY IREG IZONE ISTRAT ISUB ISEG	card image	1 N
	card image	2 N
	card image	n N

Linkage:

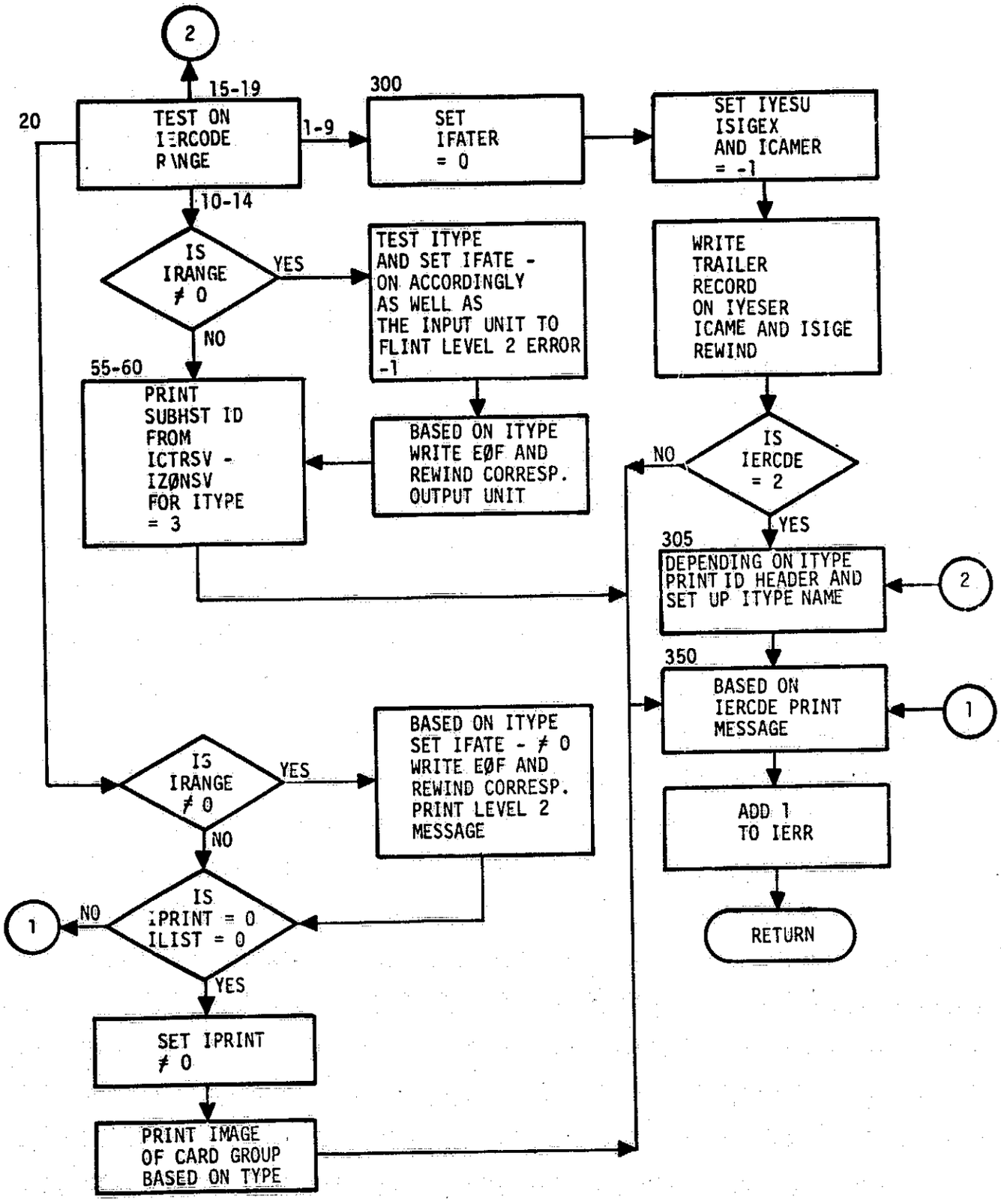
CALL ERRPRO

Subroutines Used:

None.

Processing: See flowchart.

SUBROUTINE ERRPRØ FLOWCHART



SUPPLIED UTILITY ROUTINES

Routine Day

Call Day (IYMD, IDAY)

Given IYMD (3) where

IYMD (1)	IS Day No.
IYMD (2)	IS Month No.
IYMD (3)	IS Year No.

Compute year day no. in IDAY

Routine PIMOD

Call PIMOD (A)

Convert ±A in radians to an angle 0-2π

Routine SOL (Entry ALPHA)

Call ALPHA (IFLAG)

For emphemeris usage as called by hector

computes ALPHAM and ALPHAT and IFLAG = 1

Routine PAGER (Entry Eject)

Call PAGER (NLINES)

Updates line count in NLINE with NLINES

NPAGE = 0 causes page to be restored prior to print.

NPAGE - page no.

HEADER- 80 char. 20A5

ICASE- case no.

KO - 6 print unit

INMAX is max no. of lines allowed

Initially NLINE should be set > LINMAX and NPAGE = 0

SUPPLIED UTILITY ROUTINES
(CONTINUED)

Call EJECT (NLINES)

Causes page to be restored automatically and then prints headers.

Routine CLDAY

Call CLDAY

Given IDAY-DAY no. of the year compute in LMO-the month and in LDA the day no.

Need: IYEAR = 0 - Leap Year, ≠ 0 not Leap Year

Routine KEPLER

Call KEPLER (XM, XECC, XE, ERROR)

Given XM - Mean anomaly, XECC - eccentricity

Compute: E-eccentric anomaly, error = 0 means OK

Routine LFPA

Call LFPA [FLDA, LMO, LYR, ALFGM (can be dummy), DAYS]

Given: FLDA - day of month no., LMO - month no.,
LYR - year no. compute ALFGM - right ascension and
DAYS - Zulu day no.

Routine DEGMOD

Call DEGMOND (RAD, IDEG)

Given: angle rad in radians store the angle in deg., min., sec., in IDEG(1) - (3).

Routine FZULU

Call FZULU (IOATE, IOUT)

Given Zulu date in IDATE, compute year, month and day in IOUT(1) - IOUT(3).

Routine RDMIA

Call RDMIA(FL, U)

Given double precision random no. seed in FL, comoute random no. U (0-1) based on uniform distribution.

PART V

SUBROUTINE LISTINGS

```
@ILT BLKDAT,1,760802, 58758    , 1

      BLOCK DATA                                                        BLOCK
      COMMON /FLAG/                                                     FLAG
     1ICTR5V,IREGSV,IZONSV,INP,IOUT,ICASE,NLIST,IDFRS(2),IOTOS(2),      FLAG
     2IYESU,ISIGEX,ICAMER,ICSFSH,ICTRY,IREG,IZONE,ISTRAT,ISUB,IREGP,    FLAG
     3IZONEP,ISTRAP,IFATER,IZSGC,ITYPE,IFATFY,IFATEC,IFATFS,IRANGE,     FLAG
     4ICAMR,IYESR,ISIGR,IERR,IPRINT,IFFSH,ISBHS,IERCDE,ISUBCT,          FLAG
     5IYSCT,ICMCT,ISUCT,IYSERC,ICMERC,ISIGEC,IPSEG,IYESER,ICAME,        FLAG
     6ISIGE,IREG2P,IZON2P,ISTR2P,ISUB2P,ISEG2P,INFG1P,IZON1P,ISTR1P,     FLAG
     7IREG3P,IZON3P,ILEVEL,IFILT1,IFILT2                                FLAG
     8,ILEVFS                                                           MOL3
      COMMON /SUBHIS/                                                   SUBHIS
     1NSEG,IDSEG(150),IGRP,PWHTH,AREA,PWTHT,NAGR,NAL,DELPW,DELPM,       SUBHIS
     2CV(4),NAME(2),ICSF,INXSEG                                         SUBHIS
      COMMON /YESIN/                                                    YESIN
     1ICTRY1,IREG1,IZONE1,ISTRA1,YIELDI,IERDIE(3,6),BIASYI(6),SIGYII(6),YESIN
     2YIELD,IRDTE(6),BIASY(6),SIGYIE(6)                                 YESIN
      COMMON /CAMSIN/                                                   CAMSIN
     1ICTRY2,IREG2,IZONE2,ISTRA2,ISUB2,ISEG2,PWW(4),PWM(4),PWD(4),      CAMSIN
     2BWW(4),BWM(4),BWD(4),SIGWW(4),SIGWM(4),SIGWD(4),BPW(4),SIGPW(4)   CAMSIN
      COMMON /SIGIN/                                                    SIGIN
     1ICTRY3,IREG3,IZONE3,B1W,B2W,SIG1W(6),SIG2W(6),B1M,B2M,SIG1M(6),   SIGIN
     2SIG2M(6),B1D,B2D,SIG1D(6),SIG2D(6),B1,B2,SIG1(6),SIG2(6)          SIGIN
      DATA IYESU,ICAMER,ISIGEX,INP,IOUT,ISBHS,IYESER,ICAME,ISIGE        BLOCK
     1/1,4,2,5,6,3,11,8,9/                                              BLOCK
      END                                                               BLOCK
```

```
@ ELT CAMD,1,760426, 48473      , 1

000001              SUBROUTINE CAMD                                          CAMD
000002        C     CONTROLS THE READING OF CAMS INPUT DATA GROUPS FOR ALL SEGMENTS  CAMD
000003        C     WITH = ID TO SUBHST FILE ID.                             CAMD
000004              COMMON /CAMSIN/                                          CAMSIN
000005             1ICTRY2,IREG2,IZONE2,ISTRA2,ISUB2,ISEG2,PWW(4),PWM(4),PWO(4),  CAMSIN
000006             2BWW(4),BWM(4),BWO(4),SIGWW(4),SIGWM(4),SIGWO(4),BPW(4),SIGPW(4)  CAMSIN
000007              COMMON /FLAG/                                            FLAG
000008             1ICTRSV,IREGSV,IZONSV,INP,IOUT,ICASE,ILIST,IDFFS(2),TOTUS(2),  FLAG
000009             2IYESU,ISIGEX,ICAMER,ICSFSH,ICTRY,IREG,IZONE,ISTRAT,ISUB,IREGP,  FLAG
000010             3IZONEP,ISTRAP,IFATER,IZSGC,ITYPE,IFATEY,IFATEC,IFATES,IRANGE,  FLAG
000011             4ICAMR,IYESR,ISIGN,IERR,IPRINT,IFFSB,ISBHS,IERCOE,ISUBCT,  FLAG
000012             5IYSCT,ICOCT,ISGCT,IYSERC,ICMERC,ISIGEC,IPSEG,IYESFH,ICAME,  FLAG
000013             6ISIGE,IREG2P,IZON2P,ISTR2P,ISUB2P,ISEG2P,IREG1P,IZON1P,ISTR1P,  FLAG
000014             7IREG3P,IZON3P,ILEVEL,IFILT1,IFILT2                      MOD3
000015             8,ILEVEL3                                                SUBHIS
000016              COMMON /SUBHIS/                                          SUBHIS
000017             1NSEG,IDSEG(150),IGRP,PWHTH,AREA,PWTHT,NAGR,NAL,DELPW,DELPM,  SUBHIS
000018             2CV(4),NAME(2),ICSF,IMXSEG                               SUBHIS
000019              DIMENSION INDSEG(150)                                    CAMD
000020              DATA IZ/4HZZZZ/,IZAP/0/                                  CAMD
000021              ITYPE = 2                                                CAMD
000022              DO 10 I=1,150                                            CAMD
000023              INDSEG(I) = 0                                            CAMD
000024           10 CONTINUE                                                 CAMD
000025              IF(ILEVEL .GT. IFILT1)GO TO 50                           CAMD
000026           15 IF(IFATEC .NE. 0)GO TO 500                               MOD2
000027              IF(NSEG .EQ. 0)GO TO 500                                 CAMD
000028              DO 20 K=1,150
000029              IF(IDSEG(K) .EQ. 0)GO TO 25
000030              ISEG2 = IDSEG(K)
000031              IF(ILIST .EQ. 2)WRITE(IOUT,900)ICTRY,IREG,IZONE,ISTRAT,ISUB,ISEG2,  CAMD
000032             1(PWW(I),PWM(I),PWO(I),BWW(I),SIGWW(I),BWM(I),SIGWM(I),BWO(I),  MOD3
000033             2SIGWO(I),BPW(I),SIGPW(I),I=1,4)                          CAMD
000034              WRITE(ICAME)ICTRY,IREG,IZONE,ISTRAT,ISUB,ISEG2,(PWW(I),PWM(I),  CAMD
000035             1PWO(I),BWW(I),SIGWW(I),BWM(I),SIGWM(I),BWO(I),SIGWO(I),BPW(I),  MOD3
000036             2SIGPW(I),I=1,4)                                          CAMD
000037              ICMERC = ICMERC + 1                                      CAMD
000038           20 CONTINUE                                                 CAMD
000039           25 GO TO 500                                                CAMD
000040           50 IF(ICAMR .EQ. 0)GO TO 55                                 CAMD
000041              ICAMR = 0                                                CAMD
000042              GO TO 85                                                 CAMD
000043           55 CALL SEGPWO(IND,0)                                       CAMD
000044              IF(IFATER .NE. 0)GO TO 500                               CAMD
000045              IF(ICTRY2 .EQ. IZ)GO TO 450                              CAMD
000046              IFILT1 = 0                                               CAMD
000047              IF(ISUB2 .EQ. 0)IFILT1 = 1                               CAMD
000048              IF(ISTRA2 .EQ. 0)IFILT1 = 2                              CAMD
000049              IF(IZONE2 .EQ. 0)IFILT1 = 3                              CAMD
000050              IF(IREG2 .EQ. 0)IFILT1 = 4                               CAMD
000051              IF(IFILT1 .NE. 0)ISEG2 = 0                               MOD2
000052              IF(IREG2 .LT. IREG2P)GO TO 425                           CAMD
000053              IF(IREG2 .EQ. IREG2P)GO TO 60                            CAMD
000054              GO TO 80                                                 CAMD
000055           60 IF(IZONE2 .LT. IZON2P)GO TO 425                          CAMD
000056              IF(IZONE2 .EQ. IZON2P)GO TO 65                           CAMD
000057              GO TO 80                                                 CAMD
000058           65 IF(ISTRA2 .LT. ISTR2P)GO TO 425                          CAMD
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
      IF(ISTRA2 .EQ. ISTR2P)GO TO 70                                    CAMD
      GO TO 80                                                          CAMD
   70 IF(ISUB2 .LT. ISUB2P)GO TO 425                                    CAMD
      IF(ISUB2 .EQ. ISUB2P)GO TO 75                                     LAMD
   75 IF(ISEG2 .LT. ISEG2P)GO TO 425                                    *NEW
      IF(ISEG2 .EQ. ISEG2P .AND. ISEG2 .NE. 0)GO TO 425                 *NEW
   80 IREG2P = IREG2                                                    CAMD  **-1
      IZON2P = IZONE2                                                   CAMD
      ISTR2P = ISTRA2                                                   CAMD
      ISUB2P = ISUB2                                                    CAMD
      ISEG2P = ISEG2                                                    CAMD
   85 IF(IFILT1 .EQ. 4)GO TO 200                                        CAMD
      IF(IREG - IREG2)150, 90,300                                       CAMD
   90 IF(IFILT1 .EQ. 3)GO TO 200                                        CAMD
      IF(IZONE - IZONE2)150,95,300                                      CAMD
   95 IF(IFILT1 .EQ. 2)GO TO 200                                        CAMD
      IF(ISTRAT - ISTRA2)150,100,300                                    CAMD
  100 IF(IFILT1 .EQ. 1)GO TO 200                                        CAMD
      IF(ISUB - ISUB2)150,200,300                                       CAMD
  150 ICAMR = 1                                                         CAMD
      IERCDE = 10                                                       CAMD
      IF(NSEG .NE. 0)CALL ERRPRO                                        CAMD
      GO TO 500                                                         CAMD
  200 IF(NSEG .EQ. 0)GO TO 350                                          MOD1
      IF(ISEG2 .EQ. 0)GO TO 15                                          MOD2
  205 DO 210 I=1,150                                                    CAMD
      ISAV = I                                                          CAMD
      IF(IDSEG(I) .EQ. 0)GO TO 215                                      CAMD
      IF(IDSEG(I) .EQ. ISEG2)GO TO 220                                  CAMD
  210 CONTINUE                                                          CAMD
  215 IERCDE = 15                                                       CAMD
      CALL ERRPRO                                                       MOD2
      GO TO 250                                                         CAMD
  220 INDSEG(ISAV) = 1                                                  CAMD
  222 IF(IHATER .NE. 0)GO TO 250                                        MOD2
      IF(ILIST .NE. 2)GO TO 275                                         CAMD
      WRITE(IOUT,900)ICTRY,IREG,IZONE,ISTRAT,ISUB,ISEG2,(PWW(I),PWM(I), CAMD
     1PWO(I),BWW(I),SIGWW(I),BWM(I),SIGWM(I),BWO(I),SIGWO(I),BPW(I),    MOD3
     2SIGPW(I),I=1,4)                                                   CAMD
  225 WRITE(ICAME)ICTRY,IREG,IZONE,ISTRAT,ISUB,ISEG2,(PWW(I),PWM(I),    CAMD
     1PWO(I),BWW(I),SIGWW(I),BWM(I),SIGWM(I),BWO(I),SIGWO(I),BPW(I),    MOD3
     2SIGPW(I),I=1,4)                                                   CAMD
      ICMERC = ICMERC + 1                                               CAMD
  250 CALL SEGPRO(IND,I)                                                CAMD
      IF(IHATER .NE. 0)GO TO 500                                        CAMD
      IF(IND .NE. 0)GO TO 260                                           CAMD
      IF(ISEG2 - ISEG2P)425,425,255                                     CAMD
  255 ISEG2P = ISEG2                                                    CAMD
      GO TO 205                                                         CAMD
  260 IF(ICTRY2 .EQ. IZ)GO TO 450                                       CAMD
      BACKSPACE ICAMER                                                  CAMD
      IF(ISEG2P .EQ. 0)GO TO 500                                        MOD2
      DO 270 I=1,150                                                    CAMD
      IF(IDSEG(I) .EQ. 0)GO TO 500                                      CAMD
      IF(INDSEG(I) .NE. 0)GO TO 270                                     CAMD
      IPSEG = IDSEG(I)                                                  CAMD
      IERCDE = 11                                                       CAMD
      CALL ERRPRO                                                       CAMD
  270 CONTINUE                                                          CAMD
      GO TO 500                                                         CAMD
```

```
000119          300 IERCDE = 15                                                 CAMD
000120              CALL ERRPRO                                                 CAMD
000121              CALL SEGPRO(IND,1)                                          CAMD
000122              IF(IFATER .NE. 0)GO TO 500                                  CAMD
000123              IF(IND .EQ. 0)GO TO 300                                     CAMD
000124              BACKSPACE ICAMER                                            CAMD
000125              GO TO 50                                                    CAMD
000126          350 IERCDE = 15                                                 MOD2
000127              CALL ERRPRO                                                 MOD2
000128              CALL SEGPRO(IND,1)                                          MOD2
000129              IF(IFATER .NE. 0)GO TO 500                                  MOD1
000130              IF(IND .EQ. 0)GO TO 350                                     MOD1
000131              BACKSPACE ICAMER                                            MOD1
000132              GO TO 500                                                   MOD1
000133          425 IERCDE = 2                                                  CAMD
000134              CALL ERRPRO                                                 CAMD
000135              GO TO 500                                                   CAMD
000136          450 ICAMER = -1                                                 CAMD
000137              WRITE(ICAME)IZ,(IZAP,I=1,49)                                CAMD
000138              REWIND ICAME                                                CAMD
000139          500 RETURN                                                      CAMD
000140          900 FORMAT(1H0,49X,11HCAMERR FILE/3X, 8HICTRY = ,A4, 8H IREG = ,I3,   CAMD
000141             1 9H IZONE = ,I3,10H ISTRAT = ,I4, 8H ISUB = ,I4, 8H ISEG = ,I4/   CAMD
000142             23X,95H(PWN(I),PWM(I),PWO(I),BWN(I),SIGWN(I),BWM(I),SIGWM(I),BWO(I)MOD3
000143             2,SIGWO(I),BPW(I),SIGPW(I),I=1,4) = ,4(/3X,F4.2,1X,F4.2,1X,F4.2,1X,MOD1
000144             3F6.3,1X,F6.3,1X,F6.3,1X,F4.2,1X,F4.2,1X,F4.2,1X,F6.3,1X,F4.2))    MOD1
000145              END                                                         CAMD
```

```
* ELT ERRPRO,1,760402, 58775   , 1

000001          SUBROUTINE ERRPRO                                          ERRPRO
000002    C     THIS ROUTINE CONTROLS THE PROCESSING OF ALL ERRORS,PRINTS ERROR   ERRPRO
000003    C     MESSAGES AND CARD IMAGES ACCORDING TO ERROR TYPE           ERRPRO
000004          COMMON /FLAG/                                              FLAG
000005         1ICTRSV,IREGSV,IZONSV,INP,IOUT,ICASE,ILIST,IDFWS(2),IOTUS(2),   FLAG
000006         2IYESU,ISIGEX,ICAMER,ICSESH,ICTRY,IREG,IZONE,ISTRAT,ISUB,IREGP,   FLAG
000007         3IZONEP,ISTRAP,IFATER,IZSGC,ITYPE,IFATEY,IFATEC,IFATES,IRANGE,   FLAG
000008         4ICAMR,IYESR,ISIGR,IFRR,IPRINT,IFESU,ISBHS,IERCOE,ISUBCT,   FLAG
000009         5IYSCT,ICMCT,ISGCT,IYSERC,ICMERC,ISIGEC,IPSEG,IYESER,ICAME,   FLAG
000010         6ISIGE,IREG2P,IZONE2P,ISTR2P,ISUB2P,ISEG2P,IREG1P,IZONE1P,ISTR1P,   FLAG
000011         7IREG3P,IZONE3P,ILEVEL,IFILT1,IFILT2                       FLAG
000012         8,TLEVES                                                   MOD3
000013          COMMON /CAMSIN/                                           CAMSIN
000014         1ICTRY2,IREG2,IZONE2,ISTRA2,ISUB2,ISEG2,PKW(4),PKM(4),PKO(4),   CAMSIN
000015         2BWW(4),BWM(4),BWO(4),SIGWW(4),SIGWM(4),SIGWO(4),BPW(4),SIGPW(4)   CAMSIN
000016          COMMON /STGIN/                                            SIGIN
000017         1ICTRY3,IREG3,IZONE3,B1W,B2W,SIG1W(6),SIG2W(6),B1M,B2M,SIG1M(6),   SIGIN
000018         2SIG2M(6),B1O,B2O,SIG1O(6),SIG2O(6),B1,B2,SIG1(6),SIG2(6)   SIGIN
000019          COMMON /YFSIN/                                            YFSIN
000020         1ICTRY1,IREG1,IZONE1,ISTRA1,YIELD1,IERDTE(3,6),BIASYI(6),SIGYII(6),YFSIN
000021         2YIELD,IERDTE(6),BIASY(6),SIGYIE(6)                        YFSIN
000022          DATA IZ/4HZZZZ/,IZER/0/,ICM/6HCAMERR/,IYS/6HYFSERR/,ISG/6HSTGEXT/,ERRPRO
000023         1I1,I2,I3,I4,I5,I6,I7/1,2,3,4,5,6,7/                       ERRPRO
000024          IF(ITYPE .EQ. 1)IERRP = IYS                               MOD2
000025          IF(ITYPE .EQ. 2)IERRP = ICM                               MOD2
000026          IF(ITYPE .EQ. 3)IERRP = ISG                               MOD2
000027          IF(IERCOE .GT. 19)GO TO 5                                 ERRPRO
000028          IF(IERCOE .LT. 10)GO TO 300                               ERRPRO
000029          IF(IERCOE .GE. 15 .AND. IERCOE .LE. 19)GO TO 305          ERRPRO
000030        5 IF(IRANGE .EQ. 0)GO TO 50                                 ERRPRO
000031          IF(ITYPE .EQ. 1)GO TO 20                                  ERRPRO
000032          IF(ITYPE .EQ. 2)GO TO 10                                  ERRPRO
000033          IFATES = 1                                                ERRPRO
000034          WRITE(ISIGE)IZ,(IZER,I=1,58)                              ERRPRO
000035          REWIND ISIGE                                              ERRPRO
000036          GO TO 45                                                  ERRPRO
000037       10 IFATEC = 1                                                ERRPRO
000038          WRITE(ICAME)IZ,(IZER,I=1,89)                              ERRPRO
000039          REWIND ICAME                                              ERRPRO
000040          GO TO 45                                                  ERRPRO
000041       20 CONTINUE                                                  MOD2
000042          IFATEY = 1                                                ERRPRO
000043          WRITE(IYESER)IZ,(IZER,I=1,22)                             ERRPRO
000044          REWIND IYESER                                             ERRPRO
000045       45 WRITE(IOUT,900)IERRP                                      ERRPRO
000046       50 IF(IERCOE .GT.19)GO TO 100                                ERRPRO
000047          IF(ITYPE .NE. 3)GO TO 55                                  ERRPRO
000048          WRITE(IOUT,901)ICTRSV,IREGSV,IZONSV                       ERRPRO
000049          GO TO 60                                                  ERRPRO
000050       55 WRITE(IOUT,902)ICTRY,IREG,IZONE,ISTRAT,ISUB               ERRPRO
000051       60 GO TO 350                                                 ERRPRO
000052      100 IF(IPRINT .NE. 0)GO TO 350                                ERRPRO
000053          IPRINT =1                                                 ERRPRO
000054          IF(ILIST .NE. 0)GO TO 350                                 ERRPRO
000055          IF(ITYPE .EQ. 1)GO TO 120                                 ERRPRO
000056          IF(ITYPE .EQ. 2)GO TO 110                                 ERRPRO
000057          WRITE(IOUT,903)                                           ERRPRO
000058          WRITE(IOUT,904)ICTRY3,IREG3,IZONE3,B1W,B2W,(SIG1W(I),I=1,6),(SIG2WERRPRO
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
000059      1(I),I=1,6),I1,I7                                                 ERRPRO
000060      WRITE(IOUT,905)B10,B20,(SIG10(I),I=1,6),(SIG20(I),I=1,6),I3,I7    ERRPRO
000061      WRITE(IOUT,905)B1,B2,(SIG1(I),I=1,6),(SIG2(I),I=1,6),I4,I7        ERRPRO
000062      GO TO 350                                                         ERRPRO
000063  110 WRITE(IOUT,906)                                                   ERRPRO
000064      WRITE(IOUT,907)ISEG2,ICTRY2,IREG2,IZONE2,ISTRA2,ISUB2,(PWM(I),    ERRPRO
000065     1I=1,4),(BWM(I),I=1,4),(SIGWM(I),I=1,4),I1,I6                      ERRPRO
000066      WRITE(IOUT,908)(PWM(I),I=1,4),(BWM(I),I=1,4),(SIGWM(I),I=1,4),I2, ERRPRO
000067     1I6                                                                ERRPRO
000068      WRITE(IOUT,908)(PWO(I),I=1,4),(BWO(I),I=1,4),(SIGWO(I),I=1,4),I3, ERRPRO
000069     1I6                                                                ERRPRO
000070      WRITE(IOUT,909)(SIGPW(I),I=1,4),(BPW(I),I=1,4),I4,I6              ERRPRO
000071      GO TO 350                                                         ERRPRO
000072  120 WRITE(IOUT,910)                                                   ERRPRO
000073      WRITE(IOUT,911)ICTRY1,IREG1,IZONE1,ISTRA1,YTELOT,((TEROTE(I,J),I= ERRPRO
000074     1I,3),J=1,6),I1,I5                                                 ERRPRO
000075      WRITE(IOUT,912)(BIASYI(I),I=1,6),(SIGYII(I),I=1,6),I2,I5          ERRPRO
000076      GO TO 350                                                         ERRPRO
000077  300 IFATER = 1                                                        ERRPRO
000078      IYESU =-1                                                         ERRPRO
000079      ICAMEF =-1                                                        ERRPRO
000080      ISIGEX =-1                                                        ERRPRO
000081      WRITE(IYESFR)IZ,(IZER,I=1,22)                                     ERRPRO
000082      WRITE(ICAME)IZ,(IZER,I=1,49)                                      ERRPRO
000083      WRITE(ISIGE)IZ,(IZER,I=1,58)                                      ERRPRO
000084      REWIND IYESFR                                                     MOD1
000085      REWIND ISIGE                                                      ERRPRO
000086      REWIND ICAME                                                      ERRPRO
000087      IF(IERCDE .NE. 2)GO TO 350                                        ERRPRO
000088  305 IF(ITYPE .EQ. 1)GO TO 320                                         ERRPRO
000089      IF(ITYPE .EQ. 2)GO TO 310                                         ERRPRO
000090      WRITE(IOUT,901)ICTRY3,IREG3,IZONE3                                ERRPRO
000091      GO TO 350                                                         ERRPRO
000092  310 WRITE(IOUT,913)ICTRY2,IREG2,IZONE2,ISTRA2,ISUB2,ISEG2             ERRPRO
000093      GO TO 350                                                         ERRPRO
000094  320 WRITE(IOUT,914)ICTRY1,IREG1,IZONE1,ISTRA1                         ERRPRO
000095  350 GO TO(400,405,410,500,500,500,500,500,500,415,420,500,500,500,425,ERRPRO
000096     1500,500,500,500,430,435,440,445,450,455,460,465,470,475), IERCDE  MOD3
000097  400 WRITE(IOUT,915)                                                   ERRPRO
000098      GO TO 600                                                         ERRPRO
000099  405 WRITE(IOUT,916)IERRP                                              ERRPRO
000100      GO TO 600                                                         ERRPRO
000101  410 WRITE(IOUT,917)                                                   ERRPRO
000102      GO TO 600                                                         ERRPRO
000103  415 WRITE(IOUT,918)IERRP                                              ERRPRO
000104      GO TO 600                                                         ERRPRO
000105  420 WRITE(IOUT,919)IPSEG                                              ERRPRO
000106      GO TO 600                                                         ERRPRO
000107  425 WRITE(IOUT,920)IERRP                                              ERRPRO
000108      GO TO 600                                                         ERRPRO
000109  430 WRITE(IOUT,921)                                                   ERRPRO
000110      GO TO 600                                                         ERRPRO
000111  435 WRITE(IOUT,922)                                                   ERRPRO
000112      GO TO 600                                                         ERRPRO
000113  440 WRITE(IOUT,923)                                                   ERRPRO
000114      GO TO 600                                                         ERRPRO
000115  445 WRITE(IOUT,924)                                                   ERRPRO
000116      GO TO 600                                                         ERRPRO
000117  450 WRITE(IOUT,925)                                                   ERRPRO
000118      GO TO 600                                                         ERRPRO
```

```
000119          455 WRITE(IOUT,926)                                                 ERRPRO
000120              GO TO 600                                                       ERRPRO
000121          460 WRITE(IOUT,927)                                                 ERRPRO
000122              GO TO 600                                                       ERRPRO
000123          465 WRITE(IOUT,928)                                                 ERRPRO
000124              GO TO 600                                                       ERRPRO
000125          470 WRITE(IOUT,929)                                                 ERRPRO
000126              GO TO 600                                                       ERRPRO
000127          475 WRITE(IOUT,931)                                                 MOD3
000128              GO TO 600                                                       MOD3
000129          500 WRITE(IOUT,930)                                                 ERRPRO
000130              STOP                                                            ERRPRO
000131          600 IERR = IERR + 1                                                 ERRPRO
000132              RETURN                                                          ERRPRO
000133          900 FORMAT(1H0,20X,25HLEVEL 2 ERROR,WRITING OF ,A6,13H FILE STOPPED) ERRPRO
000134          901 FORMAT(1H0,2X,6HICTRY ,A4,6H IREG ,I3,7H IZONE ,I3)             ERRPRO
000135          902 FORMAT(1H0,2X,6HICTRY ,A4,6H IREG ,I3,7H IZONE ,I3,7H ISTRAT ,I4, ERRPRO
000136             16H ISUB ,I4)                                                    ERRPRO
000137          903 FORMAT(1H0,45X,18HSTG EXT DATA GROUP)                           ERRPRO
000138          904 FORMAT(1H ,14X,A4,I3,I3,2F6.3,12F4.2,4X,2I1)                    ERRPRO
000139          905 FORMAT(1H ,24X,2F6.3,12F4.2,4X,2I1)                             ERRPRO
000140          906 FORMAT(1H0,47X,15HCAMS DATA GROUP)                              ERRPRO
000141          907 FORMAT(1H ,14X,I4,A4,I3,I3,I4,I4,4F4.2,4F6.3,4F4.2,2I1)         ERRPRO
000142          908 FORMAT(1H ,36X,4F4.2,4F6.3,4F4.2,2I1)                           ERRPRO
000143          909 FORMAT(1H ,36X,4F4.2,4F6.3,16X,2I1)                             ERRPRO
000144          910 FORMAT(1H0,47X,14HYES DATA GROUP)                               ERRPRO
000145          911 FORMAT(1H ,14X,A4,I3,I3,I4,F5.2,18I2,19X,2I1)                   ERRPRO
000146          912 FORMAT(1H ,28X,6F5.1,6F5.2,2I1)                                 MOD3
000147          913 FORMAT(1H ,2X,6HICTRY ,A4,6H IREG ,I3,7H IZONE ,I3,7H ISTRAT ,I4, ERRPRO
000148             16H ISUB ,I4,6H ISEG ,I4)                                        ERRPRO
000149          914 FORMAT(1H ,2X,6HICTRY ,A4,6H IREG ,I3,7H IZONE ,I3,7H ISTRAT ,I4) ERRPRO
000150          915 FORMAT(1H0,107H*** EITHER THE SUBSTRATA HIST. FILE HAS NOT BEEN MOERRPRO
000151             1UNTED OR IT HAS THE WRONG CASE NO. PROCESSING IS STOPPED)       ERRPRO
000152          916 FORMAT(1H0,  8H*** THE ,A6,57H INPUT DATA SET IS OUT OF ORDER. PROERRPRO
000153             1CESSING IS TERMINATED)                                          ERRPRO
000154          917 FORMAT(1H0,44H*** ID IN IDIOS IS NOT GREATER OR = TO IDFRS)     ERRPRO
000155          918 FORMAT(1H0,4H*** ,A6,69H INPUT DATA SET DOES NOT HAVE A RECORD WHIERRPRO
000156             1CH CORRESPONDS TO SUBHST ID)                                    ERRPRO
000157          919 FORMAT(1H0,52H*** CAMS INPUT DATA GROUP DOES NOT EXIST FOR SEGMENTERRPRO
000158             1 ,I4,17H IN SUBHST RECORD)                                      ERRPRO
000159          920 FORMAT(1H0, 8H*** THE ,A6,67H INPUT DATA GROUP DOES NOT MATCH WITHERRPRO
000160             1 SUBHST FILE AND IS EXTRA DATA)                                 ERRPRO
000161          921 FORMAT(1H0,49H*** IFRDTF ARRAY HAS DATES NOT IN ASCENDING ORDER) ERRPRO
000162          922 FORMAT(1H0,36H*** YTEIDT IS NOT BETWEEN 0 AND 99.9)             ERRPRO
000163          923 FORMAT(1H0,50H*** ABSOLUTE VALUE OF BIASYI IS NOT LESS THAN 99.9) ERRPRO
000164          924 FORMAT(1H0,36H*** SIGYII IS NOT BETWEEN 0 AND 99.9)             ERRPRO
000165          925 FORMAT(1H0,30H*** PW- IS NOT BETWEEN 0 AND 1)                   ERRPRO
000166          926 FORMAT(1H0,54H*** ABSOLUTE VALUE OF BW- OR BPW IS NOT LESS THAN 9.ERRPRO
000167             199)                                                             ERRPRO
000168          927 FORMAT(1H0,44H*** SIGW- OR SIGPW IS NOT BETWEEN 0 AND 9.99)     ERRPRO
000169          928 FORMAT(1H0,54H*** ABSOLUTE VALUE OF B1- OR B2- IS NOT LESS THAN 9.ERRPRO
000170             199)                                                             ERRPRO
000171          929 FORMAT(1H0,44H*** SIG1- OR SIG2- IS NOT BETWEEN 0 AND 9.99)     ERRPRO
000172          930 FORMAT(1H0,33H*** BAD ERROR CODE. JOB ABANDONED)                ERRPRO
000173          931 FORMAT(1H0,51H*** IFRDTF ARRAY HAS A BAD YEAR,MONTH OR DAY NUMBER)MOD3
000174              END                                                             ERRPRO
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
• FLT LFPA,1,760402, 58777      , 1

000001          SUBROUTINE LFPA(FLDA,LMO,LYR,DAYS)                                  LFPA
000002    C                                                                         LFPA
000003    C MODULE - P1LFPA                                                         LFPA
000004    C AUGUST 27, 1975                                                         LFPA
000005    C                                                                         LFPA
000006    C                                                                         LFPA
000007    C  JUNE 26,1973                                                           LFPA
000008    C                                                                         LFPA
000009    C  MODULE PPLFPA - POINT TARGET PERFORMANCE PREDICTOR                     LFPA
000010    C                                                                         LFPA
000011    C  GIVEN  DAY,MONTH,YEAR - SUBR. LFPA RETURNS THE RIGHT ASCENSION OF      LFPA
000012    C  GREENWICH AT MIDNIGHT OF A GIVEN DAY                                   LFPA
000013    C                                                                         LFPA
000014    C     REFERENCE EPOCH IS 0 HOUR 1JAN1950                                  LFPA
000015    C                                                                         LFPA
000016    C     COMPUTE DAYS IN FULL YEARS FROM EPOCH TO LYR                        LFPA
000017    C     DAYS TO 1JAN1963 IS 4748.                                           LFPA
000018    C                                                                         LFPA
000019          DAYS=4748.                                                          LFPA
000020    C     NOTE--LYR MUST BE GREATER THAN OR EQUAL TO (19)64                   LFPA
000021          LASTYR=LYR-1                                                        LFPA
000022          DO 20 I=63,LASTYR                                                   LFPA
000023          KREMD=MOD((I-60),4)                                                 LFPA
000024          IF(KREMD.GT.0) GO TO 10                                             LFPA
000025          DAYS=DAYS+366                                                       LFPA
000026          GO TO 20                                                            LFPA
000027       10 DAYS=DAYS+365                                                       LFPA
000028       20 CONTINUE                                                            LFPA
000029          IF(LMO-1)30,40,50                                                   LFPA
000030       30 IF(LMO-2)60,50,60                                                   LFPA
000031       40 DAYS=DAYS+FLDA-1.                                                   LFPA
000032          GO TO 270                                                           LFPA
000033       50 DAYS=DAYS+FLDA+30.                                                  LFPA
000034          GO TO 270                                                           LFPA
000035       60 KDEL=MOD((LYR-60),4)                                                LFPA
000036          IF(KDEL.GT.0) GO TO 70                                              LFPA
000037          DAYS=DAYS+59.                                                       LFPA
000038          GO TO 80                                                            LFPA
000039       70 DAYS=DAYS+58.                                                       LFPA
000040       80 IF(LMO-3) 270,170,90                                                LFPA
000041       90 IF(LMO-5) 180,190,100                                               LFPA
000042      100 IF(LMO-7) 200,210,110                                               LFPA
000043      110 IF(LMO-9) 220,230,120                                               LFPA
000044      120 IF(LMO-11)240,250,260                                               LFPA
000045      170 DAYS=DAYS+FLDA                                                      LFPA
000046          GO TO 270                                                           LFPA
000047      180 DAYS=DAYS+FLDA+31.                                                  LFPA
000048          GO TO 270                                                           LFPA
000049      190 DAYS=DAYS+FLDA+61.                                                  LFPA
000050          GO TO 270                                                           LFPA
000051      200 DAYS=DAYS+FLDA+92.                                                  LFPA
000052          GO TO 270                                                           LFPA
000053      210 DAYS=DAYS+FLDA+122.                                                 LFPA
000054          GO TO 270                                                           LFPA
000055      220 DAYS=DAYS+FLDA+153.                                                 LFPA
000056          GO TO 270                                                           LFPA
000057      230 DAYS=DAYS+FLDA+184.                                                 LFPA
000058          GO TO 270                                                           LFPA
```

```
000059          240 DAYS=DAYS+FLDA+214.                                  LFPA
000060              GO TO 270                                            LFPA
000061          250 DAYS=DAYS+FLDA+245.                                  LFPA
000062              GO TO 270                                            LFPA
000063          260 DAYS=DAYS+FLDA+275.                                  LFPA
000064          270 CONTINUE                                             LFPA
000065        C                                                          LFPA
000066              RETURN                                               LFPA
000067        C                                                          LFPA
000068              END                                                  LFPA
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
@ FLT SEE,1,760426, 48475      , 1

      C     PROGRAM SEE(ITEST=401,OTEST=401,TAPE5=ITEST,TAPE6=OTEST,SIGEXT=100SEE
      C    12,SUBHST=1002,YESERR=1002,CAMERR=1002,SIGECD=401,CAMECD=401,YESECDSEE
      C    2=401,TAPE8=CAMERR,TAPE9=SIGEXT,TAPE11=YESERR,TAPE3=SUBHST,TAPE1=  MOD1
      C    3YESECD,TAPE2=SIGECD,TAPE4=CAMECD)                                 SEE
           COMMON /FLAG/                                                      FLAG
           1ICTRSV,IREGSV,IZONSV,INP,IOUT,ICASE,ILIST,IDERS(2),IDTOS(2),      FLAG
           2IYESU,ISIGEY,ICAMER,ICSESH,ICTRY,IREG,IZONE,ISTRAT,ISUB,IPEGP,    FLAG
           3IZONEP,ISTRAP,IFATLR,IZSGC,ITYPE,IFATEY,IFATEC,IFATES,IRANGE,     FLAG
           4ICAMR,IYESR,ISIGR,IERR,IPRINT,IFESB,ISUBS,IERCDF,ISUBCI,          FLAG
           5IYSCI,ICMCI,ISGCI,IYSERC,ICMERC,ISIGEC,IPSEG,IYESER,ICAME,        FLAG
           6ISIGE,IREG2P,IZON2P,ISTR2P,ISUB2P,ISEG2P,IREG1P,IZON1P,ISTR1P,    FLAG
           7IREG3P,IZON3P,ILEVEL,IFILT1,IFILT2                                FLAG
           8,ILEVES                                                           MOD3
           COMMON /SUBHIS/                                                    SUBHIS
           1NSEG,IDSEG(150),IGRP,PWHIH,AREA,PWIHI,NAGR,NAL,DELPW,DELPM,       SUBHIS
           2CV(4),NAME(2),ICSE,IRXSEG                                         SUBHIS
           COMMON /CAMSIN/                                                    CAMSIN
           1ICTRY2,IREG2,IZONE2,ISTRA2,ISUB2,ISEG2,PWW(4),PWM(4),PWO(4),      CAMSIN
           2BWW(4),BWM(4),BWO(4),SIGWW(4),SIGWM(4),SIGWO(4),BPW(4),SIGPW(4)   CAMSIN
           COMMON /SIGIN/                                                     SIGIN
           1ICTRY3,IREG3,IZONE3,B1W,B2W,SIG1W(6),SIG2W(6),B1M,B2M,SIG1M(6),   SIGIN
           2SIG2M(6),B1O,B2O,SIG1O(6),SIG2O(6),B1,B2,SIG1(6),SIG2(6)           SIGIN
           COMMON /YESIN/                                                     YESIN
           1ICTRY1,IREG1,IZONE1,ISTRA1,YIELDI,IERDIF(3,6),BIASYI(6),SIGYII(6),YESIN
           2YIELD,IRDIF(6),BIASY(6),SIGYIL(6)                                 YESIN
           DIMENSION NAMI(2),NAMCAM(2),NAMYES(2),NAMSIG(2),IDAT(4),IDAT1(3)   SEE
           DATA NAMCAM/4HCAMS,4H ERR/,NAMYES/4HYESE,4HRROR/,NAMSIG/4HSIGS,    SEE
           14HXTEN/,IDAT/6H      ,6HYESERR,6HCAMERR,6HSIGEXT/                 SEE
           DATA IPAD/07777777777/,IZER/0/,NAMI/4HSUB ,4HHIST/                 *NEW
           IFATER = 0                                                         SEE  **-1
           IFATEY = 0                                                         SEE
           IFATEC = 0                                                         SEE
           ISIGR = 0                                                          SEE
           IFATES = 0                                                         SEE
           ICAMR = 0                                                          SEE
           IYESR = 0                                                          SEE
           IFILT1 = 0                                                         SEE
           IFILT2 = 0                                                         SEE
           IPRINT = 0                                                         SEE
           IFESB = 0                                                          SEE
           ITYPE = 0                                                          SEE
           IERR = 0                                                           SEE
           IZSGC = 0                                                          SEE
           ISUBCI = 0                                                         SEE
           IYSCI = 0                                                          SEE
           ICMCI = 0                                                          SEE
           ISGCI = 0                                                          SEE
           IYSERC = 0                                                         SEE
           ICMERC = 0                                                         SEE
           ISIGEC = 0                                                         SEE
           IPEGP = 0                                                          SEE
           IZONEP = 0                                                         SEE
           ISTRAP = 0                                                         SEE
           IREG2P = 0                                                         SEE
           IZON2P = 0                                                         SEE
           ISTR2P = 0                                                         SEE
           ISUB2P = 0                                                         SEE
           ISEG2P = 0                                                         SEE
```

```
000059          IREG1P = 0                                                  SFE
000060          IZONIP = 0                                                  SFE
000061          ISIRIP = 0                                                  SFE
000062          IREG3P = 0                                                  SFE
000063          IZONSP = 0                                                  SFE
000064          READ(INP,900)ICASE,ILIST,(IDFRS(I),I=1,2),IDTOS(1),IDTOS(2),IYT,  SFE
000065         1IST,ICTE,ICSESH                                             SFE
000066          WRITE(IOUT,901)ICASE,ILIST,(IDFRS(I),I=1,2),IDTOS(1),IDTOS(2),IYT,SFE
000067         1IST,ICTE,ICSESH                                             SFE
000068          IF(IYT .NE. 0)IYESU = IYT                                   SFE
000069          IF(IST .NE. 0)ISIGEX = IST                                  SFE
000070          IF(ICTE .NE. 0)ICAMER = ICTE                                SFE
000071          IF(IDFRS(1) .NE. 0 .OR. IDTOS(1) .NE. 0)GO TO 10            SFE
000072          IDTOS(1) = IPAD                                             SFE
000073          IDTOS(2) = IPAD                                             SFE
000074       10 IF(IDTOS(1) .LT. IDFRS(1))GO TO 20                          SFE
000075          IF(IDTOS(1) .NE. IDFRS(1))GO TO 25                                  *NEW
000076          IF(IDTOS(2) .LT. IDFRS(2))GO TO 20                          SFE
000077          GO TO 25                                                    SFE
000078       20 IERCOD = 3                                                  SFE
000079          CALL ERRPRO                                                 SFE
000080          GO TO 50                                                    SFE
000081       25 REWIND ISPHS                                                SFE
000082          IF(IYESU .GT. 0)REWIND IYESU                                MOD2
000083          IF(ISIGEX .GT. 0)REWIND ISIGEX                              MOD2
000084          IF(ICAMER .GT. 0)REWIND ICAMER                              MOD2
000085          READ(ISPHS)NAME ,      ICSE,  IMXSEG                        SFE
000086          IF(ICSESH .EQ. 0)GO TO 27                                   SFE
000087          IF(ICSESH.NE. ICSE  )GO TO 30                               SFE
000088       27 IF(NAME(1).NE.NAMT(1) .OR.NAME(2).NE. NAMT(2))GO TO 30      SFE
000089          GO TO 35                                                    SFE
000090       30 IERCOD = 1                                                  SFE
000091          CALL ERRPRO                                                 SFE
000092          GO TO 50                                                    SFE
000093       35 IF(IYESU .GT. 0)WRITE(IYESER)NAMYES,ICASE,(IZER,I=1,20)     MOD1
000094          IF(ISIGEX .GT. 0)WRITE(ISIGE)NAMSIG,ICASE,(IZER,I=1,56)     MOD1
000095          IF(ICAMER .GT. 0)WRITE(ICAME)NAMCAM,ICASE,(IZER,I=1,47)     MOD1
000096          CALL SECON                                                  SFE
000097          WRITE(IOUT,902)ISUBCT,IYSCI,ICHCT,ISGCI,IYSERC,ICASE,ICMERC,ICASE,SFE
000098         1ISIGEC,ICASE,IERR                                           SFE
000099       50 IF(IFATER .NE. 0)WRITE(IOUT,903)                            SFE
000100          IF(IFATER .EQ. 0)WRITE(IOUT,904)                            SFE
000101          IF(IFATEY .NE. 0)GO TO 60                                   SFE
000102          IF(IFATEC .NE. 0)GO TO 60                                   SFE
000103          IF(IFATES .NE. 0)GO TO 60                                   SFE
000104       55 STOP                                                        SFE
000105       60 DO 65 I=1,3                                                 SFE
000106          IDAT1(I) = IDAT(I)                                          SFE
000107       65 CONTINUE                                                    SFE
000108          IF(IFATEY .NE. 0)IDAT1(1) = IDAT(2)                         SFE
000109          IF(IFATEC .NE. 0)IDAT1(2) = IDAT(3)                         SFE
000110          IF(IFATES .NE. 0)IDAT1(3)= IDAT(4)                          SFE
000111          WRITE(IOUT,905)(IDAT1(I),I=1,3)                             SFE
000112          GO TO 55                                                    SFE
000113      900 FORMAT(I4,I1,4I3,3I2,I4)                                    SFE
000114      901 FORMAT(1H1,2X,16A,16HTOFRS      IDTOS/5X, 6HICASE  ILIST  REG  ZONMOD1
000115         1GF  REG  ZONE  IYESU  ISIGEX  ICAMR  ICSESH/4X,I4,3X,I2,4X,I3,3X, MOD1
000116         1I3,2X,I3,3X,I3,3X,I2,6X,I2,5X,I2,5X,I4)                     SFE
000117      902 FORMAT(1H1,44X,22HGEE STATUS INFORMATION/                   SFE
000118         13X,I4,5HD SUBSIPATA HISTORICAL FILE RECORDS READ/           SFE
```

```
000119          23X,I4,24H YES DATA CARD SETS READ/                          MOD3
000120          33X,I4,31H CAMS ERROR DATA CARD SETS READ/                   MOD3
000121          43X,I4,40H SIGNATURE EXTENSION DATA CARD SETS READ/          MOD3
000122          43X,I4,42H RECORDS WRITTEN ON YESERR FILE. ICSEYM = ,I4/ .   SEE
000123          53X,I4,42H RECORDS WRITTEN ON CAMERR FILE. ICSECF = ,I4/     SEE
000124          63X,I4,42H RECORDS WRITTEN ON SIGEXT FILE. ICSESF = ,I4/     SEE
000125          73X,I4,22H INPUT ERRORS DETECTED)                            SEE
000126      903 FORMAT(1H1,2X,17HJOB IS INCOMPLETE)                          SEE
000127      904 FORMAT(1H1,2X,13HJOB COMPLETED)                              SEE
000128      905 FORMAT(1H0,2X,60HFATAL ERROR OCCURRED IN SEE. FOLLOWING FILES MAY SEE
000129         1NOT BE ANY GOOD/14X,3(A6,2X))                                SEE
000130          END                                                          SEE
```

```
* FLT SEECON,1,760426, 46476    , 1

000001          SUBROUTINE SEECON                                       SEECON
000002    C     READS THE SUBJST FILE AND CONTROLS THE READING AND PROCESSING OF  SEECON
000003    C     THE 3 INPUT FILES.                                      SEECON
000004          COMMON /FLAG/                                           FLAG
000005         1ICTRSV,IREGSV,IZONSV,INP,IOUT,ICASE,ILIST,IDFRS(2),IDTOS(2),  FLAG
000006         2IYESU,ISIGEX,ICAMER,ICSFSH,ICTRY,IREG,IZONE,ISTRAT,ISUB,IRLGP,  FLAG
000007         3IZONEP,ISTRAP,IFATEK,IZSGC,ITYPE,IFATEY,IFATEC,IFATFS,IRANGE,  FLAG
000008         4ICAMR,IYESR,ISIGR,IERR,IPRINT,IEESD,ISUBS,IERCDF,ISUBCT,  FLAG
000009         5IYSCT,ICNCT,ISGCT,IYSERC,ICMERC,ISIGEC,IPSEC,IYESFH,ICAMC,  FLAG
000010         6ISIGE,IREG2P,IZON2P,ISTR2P,ISUB2P,ISEG2P,IREG1P,IZON1P,ISTR1P,  FLAG
000011         7IREG3P,IZON3P,ILEVEL,IFILT1,IFILT2                     FLAG
000012         8,ILEVES                                                MOD3
000013          COMMON /SIGIN/                                         SIGIN
000014         1ICTRY3,IREG3,IZONE3,B1W,B2W,SIG1W(6),SIG2W(6),B1M,B2M,SIG1M(6),  SIGIN
000015         2SIG2M(6),B10,B20,SIG10(6),SIG20(6),B1,B2,SIG1(6),SIG2(6)  SIGIN
000016          COMMON /YESIN/                                         YESIN
000017         1ICTRY1,IREG1,IZONE1,ISTRA1,YIELD1,IERDIF(3,6),BIASYI(6),SIGYII(6),  YESIN
000018         2YIELD,IRDIF(6),BIASY(6),SIGYIE(6)                      YESIN
000019          COMMON /CAMSIN/                                        CAMSIN
000020         1ICTRY2,IREG2,IZONE2,ISTRA2,ISUB2,ISEG2,PWW(4),PWM(4),PWO(4),  CAMSIN
000021         2BWW(4),BWM(4),BWO(4),SIGWW(4),SIGWM(4),SIGWO(4),BPW(4),SIGPW(4)  CAMSIN
000022          COMMON /SUBHIS/                                        SUBHIS
000023         1NSEG,IDSEG(150),IGRP,PWHTH,AREA,PWTHT,NAGR,NAL,DELPW,DELPM,  SUBHIS
000024         2CV(4),NAME(2),ICSE,IMXSEG                              SUBHIS
000025          DATA I//4HZZZZ//,IPAD/  0777777777/                    SEECON
000026          IFIRST = 1                                             SEECON
000027       10 IF(IEESD .NE. 0)GO TO 35                               SEECON
000028          READ(ISUBS)ICTRY,IREG,IZONE,ISTRAT,ISUB,NSEG,(IDSEG(I),I=1,IMXSEG)SEECON
000029         1,IGRP,PWHTH,AREA,PWTHT,NAGR,NAL,DELPW,DELPM,(CV(I),I=1,4)  SEECON
000030          IF(ICTRY .NE. IZ)GO TO 20                              SEECON
000031          IEESD = 1                                              SEECON
000032          IRLG = IPAD                                            SEECON
000033          IZONE = IPAD                                           SEECON
000034          ISTRAT = IPAD                                          SEECON
000035          ISUB = IPAD                                            SEECON
000036          IRANGE = 0                                             SEECON
000037          IZSGC = 0                                              SEECON
000038          ILEVEL = 5                                             MOD2
000039          GO TO 26                                                        *NEW
000040       20 ISUBCT = ISUBCT + 1                                    SEECON  **-1
000041          ILEVEL = 1                                             SEECON
000042          IF(ISTRAT .NE. ISTRAP)ILEVEL = 2                       SEECON
000043          IF(IZONE .NE. IZONEP)ILEVEL = 3                        SEECON
000044          IF(IREG .NE. IREGP)ILEVEL = 4                          SEECON
000045       25 IREGP = IREG                                           SEECON
000046          IZONEP = IZONE                                         SEECON
000047          ISTRAP = ISTRAT                                        SEECON
000048       26 IF(IREG .LT. IDFRS(1) .OR. IREG .GT. IDTOS(1))GO TO 30        *NEW
000049          IF(IREG .EQ. IDFRS(1))GO TO 28                                  *NEW
000050          IF(IREG .EQ. IDTOS(1))GO TO 29                                  *NEW
000051          GO TO 27                                                        *NEW
000052       28 IF(IZONE .LT. IDFRS(2))GO TO 30                                 *NEW
000053          GO TO 27                                                        *NEW
000054       29 IF(IZONE .GT. IDTOS(2))GO TO 30                                 *NEW
000055       27 IRANGE = 1                                                      *NEW
000056          GO TO 35                                               SEECON  **-3
000057       30 IRANGE = 0                                             SEECON
000058       35 IF(IYESU .LT. 0)GO TO 45                               SEECON
```

```
000059          IF(ILEVEL .LE. 1)GO TO 45                          SECON
000060          CALL YESD                                          SECON
000061       45 IF(ICAMER .LT. 0)GO TO 50                          SECON
000062          CALL CAMD                                          SECON
000063       50 IF(ISIGEX .LT. 0)GO TO 65                          SECON
000064          IF(ILEVEL .LE. 2)GO TO 65                          SECON
000065          IF(IFIRST .EQ. 0)GO TO 55                          SECON
000066          IFIRST = 0                                         SECON
000067          GO TO 65                                           SECON
000068       55 CALL SIGD                                          SECON
000069       65 IF(IYESU .GT. 0)GO TO 70                           SECON
000070          IF(ICAMER .GT. 0)GO TO 70                          SECON
000071          IF(ISIGEX .GT. 0)GO TO 70                          SECON
000072          RETURN                                             SECON
000073       70 IF(ILESS .EQ. 0)GO TO 75                           SECON
000074          IREGSV = IPAD                                      SECON
000075          IZONSV = IPAD                                      SECON
000076       75 IF(ILEVEL .GT. 2)GO TO 80                          SECON
000077          IZSEC = IZSEC + NSEG                               SECON
000078          GO TO 10                                           SECON
000079       80 IZSEC = NSEG                                       SECON
000080          ICTRSV = ICTRY                                     SECON
000081          IREGSV = IREG                                      SECON
000082          IZONSV =-IZONE                                     SECON
000083          ILEVLS = ILEVEL                                    MOD3
000084          GO TO 10                                           SECON
000085          END                                                SECON
```

```
• FLT SEGPRO,1,760402, 58773   , 1

000001          SUBROUTINE SEGPRO(IND,IOP)                                  SEGPRO
000002    C     READS 1 CAMS ERROR INPUT TO GROUP INTO CAMSIN COMMON        SEGPRO
000003          COMMON /FLAG/                                               FLAG
000004         1ICTRSV,IREGSV,IZONSV,INP,IOUT,ICASE,ILIST,IERRS(2),IDTOS(7),  FLAG
000005         2IYESN,ISIGEX,ICAMER,ICSESH,ICTRY,IREG,IZONE,ISTRAT,ISUB,IREGP,  FLAG
000006         3IZONEP,ISTRAP,IFATER,IZSGC,IIYPE,IFATEY,IFATEC,IFATES,IRANGE,  FLAG
000007         4ICAMR,IYESK,ISIGN,IERR,IPRINT,IFESB,ISUBS,IERCOF,ISUBCT,    FLAG
000008         5IYSCT,ICMCT,ISGCT,IYSERC,ICMERC,ISIGCT,IPSEG,IYESER,ICAML,  FLAG
000009         6ISIGR,IREG2P,IZON2P,ISTR2P,ISUB2P,ISEG2P,IREG1P,IZON1P,ISTR1P,  FLAG
000010         7IREG3P,IZON3P,ILEVEL,IFILT1,IFILT2                          FLAG
000011         8,ILEVES                                                     MOD3
000012          COMMON /CAMSIN/                                             CAMSIN
000013         1ICTRY2,IREG2,IZONE2,ISTRA2,ISUB2,ISEG2,PWW(4),PWH(4),PWO(4),  CAMSIN
000014         2BWW(4),BWH(4),BWO(4),SIGWW(4),SIGWH(4),SIGWO(4),BPW(4),SIGPW(4)  CAMSIN
000015          EQUIVALENCE (PW,PWW),(BP,BWW),(SIGW,SIGWW)                  SEGPRO
000016          DIMENSION  PW(12),BW(12),SIGW(12)                           SEGPRO
000017          DATA IZ/4HZZZZ/                                             SEGPRO
000018          DIMENSION PWT(4),BWT(4),SIGWT(4)                            MOD3
000019          IND = 0                                                     SEGPRO
000020          IPRINT = 0                                                  SEGPRO
000021          READ(ICAMER,900)ISEG2,ICTRY2,IREG2,IZONE2,ISTRA2,ISUB2,     SEGPRO
000022         1(PWT(I),I=1,4),(BWT(I),I=1,4),(SIGWT(I),I=1,4),ISEQ,ISET    MOD3
000023          IF(ICTRY2 .EQ. IZ)GO TO 510                                 SEGPRO
000024          IF(IREG2 .NE.IREG2P)GO TO 300                               SEGPRO
000025          IF(IZONE2 .NE. IZON2P)GO TO 300                             SEGPRO
000026          IF(ISTRA2 .NE. ISTR2P)GO TO 300                             SEGPRO
000027          IF(ISUB2  .NE. ISUB2P)GO TO 300                             SEGPRO
000028       10 DO 12 I=1,4                                                 MOD3
000029          PWW(I) = PWT(I)                                             MOD3
000030          BWW(I) = BWT(I)                                             MOD3
000031          SIGWW(I) = SIGWT(I)                                         MOD3
000032       12 CONTINUE                                                    MOD3
000033          IF(ILIST .NE. 0)WRITE(IOUT,901)ISEG2,ICTRY2,IREG2,IZONE2,ISTRA2,  MOD3
000034         1ISUB2,(PWW(I),I=1,4),(BWW(I),I=1,4),(SIGWW(I),I=1,4),ISEQ,ISET  MOD1
000035          IF(ISET .EQ. 6 .AND. ISEQ .EQ. 1)GO TO 20                   MOD1
000036       15 IERCOF = 2                                                  SEGPRO
000037          CALL ERRPRO                                                 SEGPRO
000038          GO TO 510                                                   SEGPRO
000039       20 READ(ICAMER,900)ISEGT,ICTRYT,IREGT,IZONET,ISTRT,ISUBT,      SEGPRO
000040         1(PWM(I),I=1,4),(BWM(I),I=1,4),(SIGWM(I),I=1,4),ISEQ,ISET    SEGPRO
000041          IF(ICTRYT .EQ. IZ)GO TO 15                                  SEGPRO
000042          IF(ILIST .NE. 0)WRITE(IOUT,902)(PWM(I),I=1,4),(BWM(I),I=1,4),  SEGPRO
000043         1(SIGWM(I),I=1,4),ISEQ,ISET                                  SEGPRO
000044          IF(IREGT .NE. IREG2 )GO TO 15                               SEGPRO
000045          IF(IZONET .NE. IZONE2)GO TO 15                              SEGPRO
000046          IF(ISTRT  .NE. ISTRA2)GO TO 15                              SEGPRO
000047          IF(ISUBT  .NE. ISUB2 )GO TO 15                              SEGPRO
000048          IF(ISEGT  .NE. ISEG2 )GO TO 15                              SEGPRO
000049          IF(ISET .NE. 6 .OR. ISEQ .NE. 2)GO TO 15                    SEGPRO
000050          READ(ICAMER,900)ISEGT,ICTRYT,IREGT,IZONET,ISTRT,ISUBT,      SEGPRO
000051         1 (PWO(I),I=1,4),(BWO(I),I=1,4),(SIGWO(I),I=1,4),ISEQ,ISET   SEGPRO
000052          IF(ICTRYT .EQ. IZ)GO TO 15                                  SEGPRO
000053          IF(ILIST .NE. 0)WRITE(IOUT,902)(PWO(I),I=1,4),(BWO(I),I=1,4),  SEGPRO
000054         1(SIGWO(I),I=1,4),ISEQ,ISET                                  SEGPRO
000055          IF(IREGT .NE. IREG2)GO TO 15                                SEGPRO
000056          IF(IZONET .NE. IZONE2)GO TO 15                              SEGPRO
000057          IF(ISTRT .NE. ISTRA2)GO TO 15                               SEGPRO
000058          IF(ISUBT .NE. ISUB2)GO TO 15                                SEGPRO
```

```
000059          IF(ISEGT .NE. ISEG2)GO TO 15                                    SEGPRO
000060          IF(ISET .NE. 6 .OR. ISEQ .NE. 3)GO TO 15                        SEGPRO
000061          READ(ICARDR,903)ISEGT,ICTRYT,IREGT,IZONET,ISTRT,ISUBT,          SEGPRO
000062         1(SIGPW(I),I=1,4),(BPW(I),I=1,4),ISEQ,ISET                       SEGPRO
000063          IF(ICTRYT .EQ. IZ)GO TO 15                                      SEGPRO
000064          IF(ILTST .NE. 0)WRITE(IOUT,904)(SIGPW(I),I=1,4),(BPW(I),I=1,4), SEGPRO
000065         1ISEQ,ISET                                                       SEGPRO
000066          IF(IREGT .NE. IREG2)GO TO 15                                    SEGPRO
000067          IF(IZONET.NE. IZONE2)GO TO 15                                   SEGPRO
000068          IF(ISTRT .NE. ISTRA2)GO TO 15                                   SEGPRO
000069          IF(ISUBT .NE. ISUB2)GO TO 15                                    SEGPRO
000070          IF(ISEGT .NE. ISEG2)GO TO 15                                    SEGPRO
000071          IF(ISET .NE. 6 .OR. ISEQ .NE. 4)GO TO 15                        SEGPRO
000072          ICNCT = ICNCT + 1                                               SEGPRO
000073          DO 25 I=1,12                                                    SEGPRO
000074          IF(PW(I) .LT. 0  .OR. PW(I) .GT. 1.0)GO TO 30                   SEGPRO
000075       25 CONTINUE                                                        SEGPRO
000076          GO TO 35                                                        SEGPRO
000077       30 IERCOD = 24                                                     SEGPRO
000078          CALL ERRPRO                                                     SEGPRO
000079       35 DO 40 I=1,12                                                    SEGPRO
000080          IF(BW(I) .LE. -10.0 .OR.BW(I) .GE. 10.0)GO TO 50                SEGPRO
000081       40 CONTINUE                                                        SEGPRO
000082          DO 45 I=1,4                                                     SEGPRO
000083          IF(BPW(I) .LE. -10.0 .OR. BPW(I) .GE. 10.0)GO TO 50             SEGPRO
000084       45 CONTINUE                                                        SEGPRO
000085          GO TO 55                                                        SEGPRO
000086       50 IERCOD = 25                                                     SEGPRO
000087          CALL ERRPRO                                                     SEGPRO
000088       55 DO 60 I=1,12                                                    SEGPRO
000089          IF(SIGW(I) .LT. 0 .OR. SIGW(I) .GE. 10.0)GO TO 70               SEGPRO
000090       60 CONTINUE                                                        SEGPRO
000091          DO 65 I=1,4                                                     SEGPRO
000092          IF(SIGPW(I) .LT. 0 .OR. SIGPW(I) .GE. 10.0)GO TO 70             SEGPRO
000093       65 CONTINUE                                                        SEGPRO
000094          GO TO 320                                                       SEGPRO
000095       70 IERCOD = 26                                                     SEGPRO
000096          CALL ERRPRO                                                     SEGPRO
000097          GO TO 320                                                       SEGPRO
000098      300 IF(IOP .EQ. 0)GO TO 10                                          MOD1
000099      310 IND = 1                                                         SEGPRO
000100      320 RETURN                                                          SEGPRO
000101      900 FORMAT(I4,A4,2I3,2I4,4F4.2,4F6.3,4F4.2,2I1)                     SEGPRO
000102      901 FORMAT(1H0,47X,15HCARD DATA GROUP/1H ,14X,I4,A4,2I3,2I4,4F4.2,  SEGPRO
000103         14F6.3,4F4.2,2I1)                                                SEGPRO
000104      902 FORMAT(1H ,36X,4F4.2,4F6.3,4F4.2,2I1)                           SEGPRO
000105      903 FORMAT(I4,A4,2I3,2I4,4F4.2,4F6.3,16X,2I1)                       SEGPRO
000106      904 FORMAT(1H ,36X,4F4.2,4F6.3,16X,2I1)                             SEGPRO
000107          END                                                             SEGPRO
```

```
* FLT SIGD,1,760402, 58768      , 1

000001                SUBROUTINE SIGD                                          SIGD
000002          C     THIS ROUTINE READS A SIGNATURE EXTENSION TO GROUP AND CHECKS FOR  SIGD
000003          C     ERRORS AND WRITES THE RECORD ON THE CAMERR FILE          SIGD
000004                COMMON /FLAG/                                            FLAG
000005               1ICTRSV,IREGSV,IZONSV,INP,IOUT,ICASE,NLIST,IDERS(2),IDTUS(2),  FLAG
000006               2IYESU,ISIGEX,ICAMER,ICSFSH,ICTRY,IREG,IZONE,ISTRAT,ISUB,IREGP,  FLAG
000007               3IZONEP,ISTRAP,IFATER,IZSGC,ITYPE,IFATEY,IFATEC,IFATES,IRANGE,  FLAG
000008               4ICAMR,IYESR,ISIGR,IERR,IPRINT,IERSB,ISBMS,IERCOF,ISUBCT,  FLAG
000009               5IYSCT,ICMCT,ISGCT,IYSFRC,ICNLRC,ISIGEC,IPSEC,IYESER,ICAME,  FLAG
000010               6ISIGE,IREG2P,IZON2P,ISTR2P,ISMB2P,ISEG2P,IREG1P,IZON1P,ISTR1P,  FLAG
000011               7IREG3P,IZON3P,ILEVEL,IFILT1,IFILT2                      FLAG
000012               8,ILEVES                                                 MOD3
000013                COMMON /SIGIN/                                          SIGIN
000014               1ICTRY3,IREG3,IZONE3,B1W,B2W,SIG1W(6),SIG2W(6),B1M,B2M,SIG1M(6),  SIGIN
000015               2SIG2M(6),B1D,B2D,SIG1D(6),SIG2D(6),B1,B2,SIG1(6),SIG2(6)  SIGIN
000016                DATA IZ/40ZZZZ/,IZER/0/                                  SIGD
000017                ITYPE = 3                                                SIGD
000018                IPRINT = 0                                               SIGD
000019                IF(ILEVES .GT. IFILT2)GO TO 50                           MOD3
000020             10 IF(IFATES .NE. 0)GO TO 500                               SIGD
000021                IF(TLIST .EQ. 2)WRITE(IOUT,900)ICTRSV,IREGSV,IZONSV,B1W,B2W,  SIGD
000022               1(SIG1W(I),I=1,6),(SIG2W(I),I=1,6),B1M,B2M,(SIG1M(I),I=1,6),  SIGD
000023               2(SIG2M(I),I=1,6),B1D,B2D,(SIG1D(I),I=1,6),(SIG2D(I),I=1,6),  SIGD
000024               3B1,B2,(SIG1(I),I=1,6),(SIG2(I),I=1,6)                   SIGD
000025                WRITE(ISIGE)ICTRSV,IREGSV,IZONSV,B1W,B2W,(SIG1W(I),I=1,6),  SIGD
000026               1(SIG2W(I),I=1,6),B1M,B2M,(SIG1M(I),I=1,6),(SIG2M(I),I=1,6),B1D,B2D SIGD
000027               2,(SIG1D(I),I=1,6),(SIG2D(I),I=1,6),B1,B2,(SIG1(I),I=1,6),  SIGD
000028               3(SIG2(I),I=1,6)                                         SIGD
000029                ISIGEC = ISIGEC + 1                                     SIGD
000030                GO TO 500                                               SIGD
000031             50 IF(ISTGP .EQ. 0)GO TO 65                                SIGD
000032                ISIGR = 0                                                SIGD
000033                GO TO 400                                               SIGD
000034             65 READ(ISIGEX,901)ICTRY3,IREG3,IZONE3,B1W,B2W,(SIG1W(I),I=1,6),  SIGD
000035               1(SIG2W(I),I=1,6),ISEQ,ISET                              SIGD
000036                IF(ICTRY3 .EQ. IZ)GO TO 475                              SIGD
000037                IF(TLIST .NE. 0)WRITE(IOUT,902)ICTRY3,IREG3,IZONE3,B1W,B2W,  SIGD
000038               1(SIG1W(I),I=1,6),(SIG2W(I),I=1,6),ISEQ,ISET              SIGD
000039                IF(IREG3 .GT. IREG3P)GO TO 70                            SIGD
000040                IF(IZONE3 .GT. IZON3P)GO TO 70                           SIGD
000041                GO TO 450                                                SIGD
000042             70 IFILT2 = 0                                               SIGD
000043                IF(IZONE3 .EQ. 0)IFILT2=3                                MOD2
000044                IF(IREG3 .EQ. 0)IFILT2 = 4                               SIGD
000045                IREG3P = IREG3                                           SIGD
000046                IZON3P = IZONE3                                          SIGD
000047                IF(ISET .NE. 7 .OR. ISEQ .NE. 1)GO TO 450                SIGD
000048                READ(ISIGEX,901)ICTRYT,IREGT,IZONET,B1M,B2M,(SIG1M(I),I=1,6),  SIGD
000049               1(SIG2M(I),I=1,6),ISEQ,ISET                              SIGD
000050                IF(ICTRYT .EQ. IZ)GO TO 450                              SIGD
000051                IF(TLIST .NE. 0)WRITE(IOUT,903)B1M,B2M,(SIG1M(I),I=1,6),  SIGD
000052               1(SIG2M(I),I=1,6),ISEQ,ISET                              SIGD
000053                IF(IREG3 .NE. IREGT)GO TO 450                            SIGD
000054                IF(IZONE3 .NE. IZONET)GO TO 450                          SIGD
000055                IF(ISEQ .NE. 2 .OR. ISET .NE. 7)GO TO 450                SIGD
000056                READ(ISIGEX,901)ICTRYT,IREGT,IZONET,B1D,B2D,(SIG1D(I),I=1,6),  SIGD
000057               1(SIG2D(I),I=1,6),ISEQ,ISET                              SIGD
000058                IF(ICTRYT .EQ. IZ)GO TO 450                              SIGD
```

```
      IF(ILIST .NE. 0)WRITE(IOUT,903)B10,B20,(SIG10(I),I=1,6),          SIGD
     1(SIG20(I),I=1,6),ISEQ,ISET                                        SIGD
      IF(IREG3 .NE. IREG1)GO TO 450                                     SIGD
      IF(IZONE3 .NE. IZONE1)GO TO 450                                   SIGD
      IF(ISEQ .NE. 5 .OR. ISET .NE. 7)GO TO 450                         SIGD
      READ(ISIGEX,901)ICTRYT,IREGT,IZONET,B1,B2,(SIG1(I),I=1,6),        SIGD
     1(SIG2(I),I=1,6),ISEQ,ISET                                         SIGD
      IF(ICTRY1 .EQ. IZ)GO TO 450                                       SIGD
      IF(ILIST .NE. 0)WRITE(IOUT,903) B1,B2,(SIG1(I),I=1,6),(SIG2(I),I= SIGD
     21,6),ISEQ,ISET                                                    SIGD
      IF(IREG1 .NE. IREG3)GO TO 450                                     SIGD
      IF(IZONE1 .NE. IZONE3)GO TO 450                                   SIGD
      IF(ISEQ .NE. 4 .OR. ISET .NE. 7)GO TO 450                         SIGD
      ISGCT = ISGCT + 1                                                 SIGD
      IF(B1W .LE. -10. .OR. B1W .GE. 10.)GO TO 200                      SIGD
      IF(B2W .LE. -10. .OR. B2W .GE. 10.)GO TO 200                      SIGD
      IF(B1M .LE. -10. .OR. B1M .GE. 10.)GO TO 200                      SIGD
      IF(B2M .LE. -10. .OR. B2M .GE. 10.)GO TO 200                      SIGD
      IF(B10 .LE. -10. .OR. B10 .GE. 10.)GO TO 200                      SIGD
      IF(B20 .LE. -10. .OR. B20 .GE. 10.)GO TO 200                      SIGD
      IF(B1 .LE. -10. .OR. B1 .GE. 10.)GO TO 200                        SIGD
      IF(B2 .LE. -10. .OR. B2 .GE. 10.)GO TO 200                        SIGD
      GO TO 210                                                         SIGD
  200 IERCDE =-27                                                       SIGD
      CALL ERRPRO                                                       SIGD
  210 DO 240 I=1,6                                                      SIGD
      IF(SIG1W(I) .LT. 0 .OR. SIG1W(I) .GE. 10.)GO TO 250               SIGD
      IF(SIG2W(I) .LT. 0 .OR. SIG2W(I) .GE. 10.)GO TO 250               SIGD
      IF(SIG1M(I) .LT. 0 .OR. SIG1M(I) .GE. 10.)GO TO 250               SIGD
      IF(SIG2M(I) .LT. 0 .OR. SIG2M(I) .GE. 10.)GO TO 250               SIGD
      IF(SIG10(I) .LT. 0 .OR. SIG10(I) .GE. 10.)GO TO 250               SIGD
      IF(SIG20(I) .LT. 0 .OR. SIG20(I) .GE. 10.)GO TO 250               SIGD
      IF(SIG1(I) .LT. 0 .OR. SIG1(I) .GE. 10.)GO TO 250                 SIGD
      IF(SIG2(I) .LT. 0 .OR. SIG2(I) .GE. 10.)GO TO 250                 SIGD
  240 CONTINUE                                                          SIGD
      GO TO 400                                                         SIGD
  250 IERCDE = 28                                                       SIGD
      CALL ERRPRO                                                       SIGD
  900 FORMAT(1H0,49X,11HSIGEXT FILE/3X, 8HICTRY = ,A4, 8H IREG = ,I3,   SIGD
     1 9H IZONE = ,I3/3X,6HB1W = ,F6.3,7H B2W = ,F6.3,9H SIG1W = ,6(F4.2SIGD
     2,1X),9H SIG2W = ,6(F4.2,1X)/3X,6HB1M = ,F6.3,7H B2M = ,F6.3,      SIGD
     39H SIG1M = ,6(F4.2,1X),9H SIG2M = ,6(F4.2,1X)/3X,6HB10 = ,F6.3,   SIGD
     47H B20 = ,F6.3,9H SIG10 = ,6(F4.2,1X),9H SIG20 = ,6(F4.2,1X)/3X,  SIGD
     55HB1 = ,F6.3,6H B2 = ,F6.3,8H SIG1 = ,6(F4.2,1X),8H SIG2 = ,      SIGD
     66(F4.2,1X))                                                       NODP
  901 FORMAT(4X,A4,I3,I3,2F6.3,12F4.2,4X,2I1)                           SIGD
  902 FORMAT(1H0,45X18HSIG EXT DATA GROUP/15X,4X,A4,I3,I3,2F6.3,12F4.2, SIGD
     14X,2I1)                                                           SIGD
  903 FORMAT(29X,2F6.3,12F4.2,4X,2I1)                                   SIGD
  400 IF(IREG3 .EQ. 0)GO TO 425                                         SIGD
      IF(IREGSV - IREG3)435,405,440                                     SIGD
  405 IF(IZONE3 .EQ. 0)GO TO 425                                        SIGD
      IF(IZONSV - IZONE3)435,425,440                                    SIGD
  425 GO TO 10                                                          SIGD
  435 ISIGE = 1                                                         SIGD
      IF(IZSUC .EQ. 0)GO TO 500                                         SIGD
      IERCDE = 10                                                       SIGD
      CALL ERRPRO                                                       SIGD
      GO TO 500                                                         SIGD
  440 IERCDE = 15                                                       SIGD
```

```
000119          CALL ERRPRO                                          SIGD
000120          GO TO 50                                             SIGD
000121      450 IERCOD = 2                                           SIGD
000122          CALL ERRPRO                                          SIGD
000123          GO TO 500                                            SIGD
000124      475 ISIGEX = -1                                          SIGD
000125          WRITE(ISIGE)IZ,(IZER,I=1,58)                         SIGD
000126          REWIND ISIGE                                         SIGD
000127      500 RETURN                                               SIGD
000128          END
```

```
@ FLT YESD,1,760402, 58765      , 1

000001          SUBROUTINE YESD                                                 YESD
000002    C     READS A YES DAT CARD GROUP,CHECKS THE DATA FOR ERRORS AND WRITES YESD
000003    C     A RECORD ON THE YESERR FILE                                     YESD
000004          COMMON /FLAG/                                                   FLAG
000005         1ICTRSV,IGEGSV,IZONSV,INP,IOUT,ICASE,ILIST,IDERS(2),IDTUS(2),    FLAG
000006         2IYESU,ISIGEX,ICANER,ICSESH,ICTRY,IREG,IZONE,ISTRAT,ISUB,IREGP,  FLAG
000007         3IZONEP,ISTRAP,IFATER,IZSGC,ITYPE,IFATEY,IFATEC,IFATES,IRANGE,   FLAG
000008         4ICAMR,IYESR,ISICN,IERR,IPPINT,IFESB,ISBHS,IERCDF,ISHBCI,        FLAG
000009         5IYSCI,ICMCI,ISGCI,IYSERC,ICMERC,ISIGEC,IPSEC,IYESER,ICAME,      FLAG
000010         6ISIGL,IREG2P,IZON2P,ISTR2P,ISUB2P,ISEG2P,IREG1P,IZON1P,ISTR1P,   FLAG
000011         7IREG3P,IZON3P,ILEVEL,IFILT1,IFILT2                              FLAG
000012         8,ILEVES                                                         MOD3
000013          COMMON /YESIN/                                                  YESIN
000014         1ICTRY1,IREG1,IZONE1,ISTRA1,YIELDI,IERDIE(3,6),BIASYI(6),SIGYII(6),YESIN
000015         2YILEB,ITRDTE(6),BIASY(6),SIGYIE(6)                             YESIN
000016          DIMENSION IDIEN(6)                                              YESD
000017          DATA IZ/4HZZZZ/,IZER/0/,ICON/4HUSA /,ICON1/4HCARD/              YESD
000018          ITYPE = 1                                                       YESD
000019          IF(ICTRY .EQ. ICON .OR. ICTRY .EQ. ICON1)GO TO 2                YESD
000020          XCON = 100.                                                     YESD
000021          XCOM = -XCON                                                    YESD
000022          XCONV = 4.0                                                     MOD7
000023          GO TO 5                                                         YESD
000024        2 XCON = 100.*1.4869664                                           MOD7
000025          XCOM = -XCON                                                    YESD
000026          XCONV = 1.4869664                                               MOD7
000027        5 IF(IYESR .EQ. 0)GO TO 10                                        YESD
000028          IYESR = 0                                                       YESD
000029          GO TO 200                                                       YESD
000030       10 READ(IYESU,900)ICTRY1,IREG1,IZONE1,ISTRA1,YIELDI,((IERDTE(I,J),I=1YESD
000031         1,3),J=1,6),ISEQ,ISET                                            YESD
000032          IF(ICTRY1 .EQ. IZ)GO TO 475                                     YESD
000033          IF(ILIST .NE. 0)WRITE(IOUT,901)ICTRY1,IREG1,IZONE1,ISTRA1,YIELDI, YESD
000034         1((IERDTE(I,J),I=1,3),J=1,6),ISEQ,ISET                           YESD
000035          IF(IREG1 .GT. IREG1P)GO TO 20                                   YESD
000036          IF(IZONE1 .GT. IZON1P)GO TO 20                                  YESD
000037          IF(ISTRA1 .GT. ISTR1P)GO TO 20                                  YESD
000038          IF(IZONE1 .GT. IZON1P)GO TO 20                                  YESD
000039          IF(ISTRA1 .GT. ISTR1P)GO TO 20                                  YESD
000040          GO TO 450                                                       YESD
000041       20 IF(ISEQ .NE. 1 .OR. ISET .NE.5)GO TO 450                        YESD
000042          IREG1P = IREG1                                                  YESD
000043          IZON1P = IZONE1                                                 YESD
000044          ISTR1P = ISTRA1                                                 YESD
000045          READ(IYESU,902)ICTRY1,IREG1,IZONE1,ISTRA1,(BIASYI(I),I=1,6),    YESD
000046         1(SIGYII(I),I=1,6),ISEQ,ISET                                     YESD
000047          IF(ICTRY1 .EQ. IZ)GO TO 450                                     YESD
000048          IF(ILIST .NE. 0)WRITE(IOUT,903)(BIASYI(I),I=1,6),(SIGYII(I),I=1,6)YESD
000049         1,ISEQ,ISET                                                      YESD
000050          IF(ISEQ .NE. 2 .OR. ISET .NE. 5)GO TO 450                       YESD
000051          IF(IREG1 .NE. IREG1P)GO TO 450                                  YESD
000052          IF(IZONE1 .NE. IZON1P)GO TO 450                                 YESD
000053          IF(ISTRA1 .NE. ISTR1P)GO TO 450                                 YESD
000054          IYSCI = IYSCI + 1                                               YESD
000055          IPRINT = 0                                                      YESD
000056          IF(IERDTE(1,1) .EQ. 0)GO TO 190                                 YESD
000057          DO 30 I=1,6                                                     YESD
000058          IF(IERDTE(1,I) .EQ. 0)GO TO 26                                  MOD3
```

```
      IF(IERDTE(1,I) .GT. 60)GO TO 22                                   MOD3
      GO TO 32                                                          MOD3
   22 IF(IERDTE(2,I) .GT. 0 .AND. IERDTE(2,I) .LE. 12)GO TO 24          MOD3
      GO TO 32                                                          MOD3
   24 IF(IERDTE(3,I) .GT. 0 .AND. IERDTE(3,I) .LE. 31)GO TO 26          MOD3
      GO TO 32                                                          MOD3
   26 FLDA = IERDTE(3,I)                                                MOD3
      IDTEM(I) = 0                                                      MOD1
      IF(FLDA .EQ. 0.0)GO TO 30                                         MOD1
      CALL TPPA(FLDA ,IERDTE(2,I),IERDTE(1,I),DAYS)                     YESD
      IDTEM(I)  = DAYS                                                  YESD
   30 CONTINUE                                                          YESD
      GO TO 50                                                          MOD3
   32 IERCDE = 29                                                       MOD3
      CALL ERRPRO                                                       MOD3
      GO TO 55                                                          MOD3
   34 DO 35 I=1,5                                                       MOD3
      IF(IDTEM(I+1) .EQ. 0)GO TO 55                                     MOD1
      IF(IDTEM(I) .LT. IDTEM(I+1))GO TO 35                              YESD
      GO TO 40                                                          YESD
   35 CONTINUE                                                          YESD
      GO TO 55                                                          YESD
   40 IERCDE = 20                                                       YESD
      CALL ERRPRO                                                       YESD
   55 DO 60 I=1,6                                                       YESD
      IF(SIGYII(I) .LT. 0 .OR. SIGYII(I) .GE. XCON)GO TO 65             YESD
   60 CONTINUE                                                          YESD
      GO TO 70                                                          YESD
   65 IERCDE = 23                                                       YESD
      CALL ERRPRO                                                       YESD
   70 DO 75 I=1,6                                                       YESD
      IF(BIASYI(I) .LE. XCOM .OR. BIASYI(I) .GE. XCON)GO TO 80          YESD
   75 CONTINUE                                                          YESD
      GO TO 190                                                         YESD
   80 IERCDE = 22                                                       YESD
      CALL ERRPRO                                                       YESD
  190 IF(YIELDI .LT. 0 .OR. YIELDI .GE. XCON)GO TO 195                  YESD
      GO TO 200                                                         YESD
  195 IERCDE = 21                                                       YESD
      CALL ERRPRO                                                       YESD
  200 IF(IREG - IREG1)240,205,250                                       YESD
  205 IF(IZONE - IZONE1)240,210,250                                     YESD
  210 IF(ISTRAT - ISTRAT)240,215,250                                    YESD
  215 IF(TEATEY .NE. 0)GO TO 500                                        YESD
      YIELD = YIELDI/XCONV                                              MOD7
      IF(IERDTE(1,1) .EQ. 0)GO TO 230                                   YESD
      DO 220 I=1,6                                                      YESD
      ITRDTE(I) = IDTEM(I)                                              YESD
      BIASY(I) = BIASYI(I)/XCONV                                        MOD7
      SIGYIF(I) = SIGYII(I)/XCONV                                       MOD7
  220 CONTINUE                                                          YESD
  230 IF(IDIST .EQ. 2)WRITE(IOUT,904)ICTRY,IREG,IZONE,ISTRAT,YIELD,     YESD
     1(ITRDTE(I),BIASY(I),SIGYIF(I),I=1,6)                              YESD
      WRITE(IYESER)ICTRY,IREG,IZONE,ISTRAT,YIELD,(ITRDTE(I),BIASY(I),   YESD
     1SIGYIF(I),I=1,6)                                                  YESD
      IYSERC = IYSERC + 1                                               YESD
      GO TO 500                                                         YESD
  240 IYESR = 1                                                         YESD
      IERCDE = 10                                                       YESD
      CALL ERRPRO                                                       YESD
```

```
000119          GO TO 500                                                YESD
000120      250 IERCOF = 15                                              YESD
000121          CALL ERRPRO                                              YESD
000122          GO TO 5                                                  YESD
000123      450 IERCOE = 2                                               YESD
000124          CALL ERRPRO                                              YESD
000125          GO TO 500                                                YESD
000126      475 IYESU = -1                                               YESD
000127          WRITE(IYESER)IZ,(IZER,I=1,22)                            YESD
000128          REWIND IYESER                                            YESD
000129      500 RETURN                                                   YESD
000130      900 FORMAT(4X,A4,2I3,I4,F5.2,18I2,19X,2I1)                   YESD
000131      901 FORMAT(1H0,47X,14HYES DATA GROUP/15X,4X,A4,2I3,I4,F5.2,18I2,19X,  YESD
000132         12I1)                                                     YESD
000133      902 FORMAT(4X,A4,2I3,I4,6F5.1,6F5.2,2I1)                     YESD
000134      903 FORMAT(33X,6F5.1,6F5.2,2I1)                              YESD
000135      904 FORMAT(50X,11HYESERR FILE/3X,8HICTRY = ,A4, 8H IREG =  ,I3, 9H IZOYESD
000136         1KF = ,I3, 9H ISTRAT =,I4/3X,4HY = ,F5.2,40H (ITRDTE(I),8IASY(I),SIYESD
000137         2GYIF(I),I=1,6) = /3X,6(I5,1X,F5.1,1X,F5.2,1X))              YESD
000138          END                                                      YESD
```

15. TOC